



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

**WILLIAN CARDOSO DE MORAES**

**SISTEMA PARA GESTÃO DE ACADEMIA PERSONALIZADA**

Assis

2014

## **SISTEMA PARA GESTÃO DE ACADEMIA PERSONALIZADA**

Projeto de pesquisa apresentado ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis - IMESA e Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando:** WILLIAN CARDOSO DE MORAES

**Orientador:** OSMAR APARECIDO MACHADO

Assis

2014

## FICHA CATALOGRÁFICA

MORAES, Willian Cardoso de

Sistema para Gestão de Academia Personalizada./ Willian Cardoso de Moraes. Fundação Educacional do Município de Assis – FEMA - Assis, 2014. 60 p.

Orientador: Prof. Dr. Osmar Aparecido Machado

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA/FEMA

1.Gerenciamento 2.Academia 3.Controle

CDD: 001.61

Biblioteca da FEMA

# SISTEMA PARA GESTÃO DE ACADEMIA PERSONALIZADA

WILLIAN CARDOSO DE MORAES

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Superior de Tecnologia em Análise e desenvolvimento de Sistemas, analisado pela seguinte comissão examinadora:

Orientador: Osmar Aparecido Machado

Analisador: Domingos de Carvalho Villela Junior

ASSIS

2014

## DEDICATÓRIA

Dedico este trabalho a todos que confiaram em mim e que me deram total apoio para que eu pudesse concretizar este projeto.

## AGRADECIMENTOS

Primeiramente, agradeço a Deus por me proporcionar mais esta experiência de vida, por permitir com que eu pudesse trilhar esse caminho de forma com que eu pudesse superar cada desafio que fosse considerado como obstáculo.

Agradeço ao meu professor e orientador, Professor Osmar, que não permitiu que eu sequer pensasse em desistir deste projeto. Agradeço pela total confiança e estímulo durante esse processo.

Aos meus amigos, que foram mais do que colegas de sala, o meu muito obrigado por cada momento e por cada conhecimento compartilhado. Só tenho a agradecer o apoio de cada amigo em cada etapa, tanto os amigos que estão por perto, quanto os que estão longe. Ao Renato, o meu muito obrigado. Deborah, Paula, Márcio, Tati, Fran, Barbara, Jaqueline, Bia, Iza, Jéssica e Giovanna: agradeço pelo apoio essencial. Obrigado por serem como uma escada na minha subida.

À minha família, fica o agradecimento por estarem ao meu lado em cada escolha.

## RESUMO

Este trabalho tem como objetivo o desenvolvimento de um software para gerenciar uma academia personalizada, onde os alunos serão supervisionados pelos profissionais desta área, a fim de que o desempenho de cada aluno possa estar em constante desenvolvimento.

O intuito do sistema é abranger um mercado que tem apresentado uma grande demanda nos últimos tempos, levando em consideração que o número de academias e a busca por este ramo têm crescido frequentemente.

Para a elaboração deste projeto, é fundamental o conhecimento das ferramentas que serão utilizadas desde a fase do planejamento até a fase do desenvolvimento e manutenção do sistema.

Palavras-chave: Gerenciamento; Academia; Software.

## ABSTRACT

This project has as objective the development of software to manage a personalized gym, where the students will be supervised by the professionals of this field, in order that the performance of each student can be in constant development.

The system's objective is to embrace a market that has presented a huge demand in the last years, taking in consideration that the number of gyms and the search for this area has grown frequently.

In order to elaborate this project, it is fundamental the knowledge of the tools that will be used since the planning phase until the phase of development and maintenance of the system.

Key words: Management; Gym; Software.

## LISTA DE FIGURAS

Figura 1 - EAP (Estrutura Analítica do Projeto) .....	21
Figura 2 - Diagrama de UC Geral .....	23
Figura 3 - CSU01 Efetuar acesso .....	23
Figura 4 – CSU02 Manter alunos .....	24
Figura 5 – CSU03 Manter exercícios.....	25
Figura 6 – CSU04 Manter instrutores .....	26
Figura 7– CSU05 Manter treino.....	27
Figura 8 – CSU06 Cadastrar plano de treino .....	28
Figura 9 – CSU07 Manter usuários.....	29
Figura 10– CSU08 Gerar desempenho.....	30
Figura 11– CSU09 Registrar entrada/saída .....	31
Figura 12 – CSU10 Controlar pagamentos .....	32
Figura 13 – CSU11 Emitir relatório de desempenho .....	33
Figura 14 – CSU12 Emitir plano de treino .....	34
Figura 15 – CSU13 Validar acesso.....	35
Figura 16 – CSU14 Gerar relatórios .....	36
Figura 17 – Diagrama de classes.....	38
Figura 18 – Atividade Efetuar acesso.....	39
Figura 19 – Atividade Manter alunos .....	39
Figura 20 – Atividade Manter exercícios.....	40
Figura 21 – Atividade Manter instrutores .....	40
Figura 22 - Atividades Cadastrar treinos.....	41
Figura 23 – Atividade Cadastrar planos de treino .....	41
Figura 24 – Atividade Manter usuários.....	42

Figura 25 – Atividade Cadastrar desempenhos.....	43
Figura 26 – Atividade Registrar Entrada/Saída.....	43
Figura 27 – Atividade Controlar pagamentos.....	44
Figura 28 – Atividade Gerar relatório de desempenho.....	44
Figura 29 – Atividade Gerar plano de treino.....	45
Figura 30 – Atividade Validar acesso.....	45
Figura 31 – Atividade Gerar relatórios.....	46
Figura 32 – Modelo Entidade-Relacionamento.....	47
Figura 33 – Classe aluno model.....	49
Figura 34 – Void incluir.....	49
Figura 35 – Void alterar.....	50
Figura 36 – Void excluir.....	50
Figura 37 – Camada BLL.....	51
Figura 38– Tela de Cadastro de usuário.....	52
Figura 39 – Tela de acesso.....	52
Figura 40 – Tela de verificação de acesso.....	53
Figura 41 - Tela Cadastro de Exercícios.....	53
Figura 42 - Tela Cadastro de Desempenho.....	54
Figura 43 - Tela Cadastro de Instrutores.....	54
Figura 44 - Tela Pesquisar Alunos.....	55
Figura 45 - Tela Pesquisar Exercício.....	55
Figura 46 - Tela Cadastro de Treinos.....	56

## LISTA DE TABELAS

Tabela 1 - Lista de eventos .....	22
Tabela 2 – Efetuar acesso .....	24
Tabela 3 - Manter alunos .....	25
Tabela 4 - Manter exercícios .....	26
Tabela 5 - Manter instrutores .....	27
Tabela 6 – Cadastrar treinos .....	28
Tabela 7 - Cadastrar plano de treino .....	29
Tabela 8 – Manter usuários .....	30
Tabela 9 – Gerar desempenho .....	31
Tabela 10 - Registrar entrada/saída .....	32
Tabela 11 - Controlar pagamentos .....	33
Tabela 12 - Emitir relatório de desempenho .....	34
Tabela 13 – Emitir plano de treino .....	35
Tabela 14 - Validar acesso .....	36
Tabela 15 – Gerar relatórios .....	37

## LISTA DE SIGLAS

BLL	Business Logic Layer
DAL	Data Access Layer
EAP	Estrutura Analítica do Projeto
IMC	Índice de Massa Corporal
UC	Use Case
UML	Unified Modeling Language
WBS	Work Breakdown Structure

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>15</b>
<b>2. CONCEITUALIZAÇÃO .....</b>	<b>16</b>
2. 1. CASOS DE USO .....	16
<b>2. 2. DIAGRAMA DE CLASSES .....</b>	<b>17</b>
2. 3. DIAGRAMA DE ATIVIDADES .....	18
2. 4. MODELO ENTIDADE RELACIONAMENTO .....	18
<b>3. DESENVOLVIMENTO DO PROJETO.....</b>	<b>19</b>
3. 1. OBJETIVO .....	19
3.2. PUBLICO-ALVO .....	19
3.3. JUSTIFICATIVA.....	19
3.4. FERRAMENTAS UTILIZADAS.....	20
3.5. MÉTODO DE DESENVOLVIMENTO.....	20
3. 6. CASOS DE USO .....	21
3. 6. 1. Efetuar acesso .....	23
3.6.2. Manter alunos .....	24
3. 6. 3. Manter exercícios .....	25
3. 6. 4. Manter instrutores .....	26
3. 6. 5. Cadastrar treinos .....	27
3. 6. 6. Cadastrar plano de treino .....	28
3. 6. 7. Manter usuários.....	29
3. 6. 8. Gerar desempenho .....	30
3. 6. 9. Registrar entrada/saída.....	31

3. 6. 10. Controlar pagamentos.....	32
3. 6. 11. Emitir relatório de desempenho .....	33
3. 6. 12. Emitir plano de treino .....	34
3. 6. 13. Validar acesso.....	35
3. 6. 14. Gerar relatórios .....	36
<b>3. 7. DIAGRAMA DE CLASSES .....</b>	<b>38</b>
<b>3. 8. DIAGRAMAS DE ATIVIDADES.....</b>	<b>38</b>
3. 8. 1. Atividade Efetuar Acesso.....	39
3. 8. 2. Atividade Manter alunos.....	39
3. 8. 3. Atividade Manter exercícios .....	40
3. 8. 4. Atividade Manter instrutores .....	40
3. 8. 5. Atividade Cadastrar treinos.....	41
3. 8. 6. Atividade Cadastrar planos de treino.....	41
3. 8. 7. Atividade Manter usuários .....	42
3. 8. 8. Atividade Cadastrar desempenhos.....	43
3. 8. 9. Atividade Registrar Entrada/Saída .....	43
3. 8. 10. Atividade Controlar pagamentos .....	44
3. 8. 11. Atividade Gerar relatório de desempenho.....	44
3. 8. 12. Atividade Gerar plano de treino .....	45
3. 8. 13. Atividade Validar acesso.....	45
3. 8. 14. Atividade Gerar relatórios .....	46
<b>3. 9. MODELO ENTIDADE-RELACIONAMENTO .....</b>	<b>47</b>
<b>4. IMPLEMENTAÇÃO DO SISTEMA.....</b>	<b>48</b>
4.1. CAMADA MODEL .....	48
4.2. CAMADA DAL .....	49

4.3. CAMADA BLL.....	50
4.4. CAMADA GRÁFICA.....	51
<b>5. CONCLUSÃO FINAL.....</b>	<b>57</b>
<b>REFERÊNCIAS .....</b>	<b>58</b>

## 1. INTRODUÇÃO

Não é de hoje que a necessidade de controlar e administrar os recursos de forma prática e eficaz tem se tornado uma grande meta tanto para pequenos, quanto para grandes empresários. Nas organizações, diariamente, diversas informações são processadas para garantir um gerenciamento adequado para qualquer organização e estas auxiliam nas tomadas de decisões.

Oliveira (2008) salienta que tendo em vista esses fatos, a necessidade de centralizar as informações para que elas sejam devidamente processadas está cada vez mais presente no cotidiano das pequenas e grandes empresas. Tais informações não são processadas de forma adequadas para que se obtenham grandes resultados através das mesmas.

Miranda (1991) já apresentava a grande demanda na busca no ramo de empresas que trabalham com a área de saúde e estética, e que, atualmente, tem ganhado espaço no mercado, e essa demanda faz com que a necessidade de gerenciar as informações decorrentes deste ramo seja cada vez maior. Conseqüentemente, com esse grande aumento na demanda, estes estabelecimentos necessitam de um gerenciamento adequado para garantir aos seus clientes maior retorno em relação ao investimento que lhe é aplicado. Os sistemas de informação são necessários para garantir ao usuário (ou empresários) a segurança e total controle das movimentações de suas organizações.

Por isso, este estudo, alinhado com essas demandas propõe o desenvolvimento de um sistema que visa à criação de um software para gerenciamento de academias onde a supervisão e dedicação a cada aluno é constante, a fim de atingir esta fatia da população que está em constante crescimento.

## 2. CONCEITUALIZAÇÃO

Para o desenvolvimento do projeto de software, o qual é o objetivo deste trabalho, o conhecimento das ferramentas necessárias para a execução de tal feito é de extrema importância.

A revisão da literatura foi realizada tanto em meio impresso (livros, revistas, artigos) como em meio eletrônico, em sites especializados e outros recursos disponibilizados, como as bibliotecas virtuais.

Levando em consideração a necessidade de conhecer ou até mesmo aprimorar o conhecimento já adquirido a respeito destas ferramentas de estudo, viu-se a necessidade de elaboração de pesquisas através de revisões literárias para aprofundar o conhecimento sobre essas ferramentas consideradas fundamentais no desenvolvimento de um projeto.

### 2. 1. CASOS DE USO

Durante a fase de levantamento de requisitos, onde são levantadas as principais características e necessidades do software desenvolvido, a elaboração dos casos de uso é um fator primordial. Depois de estabelecido e desenvolvido o diagrama de casos de uso (UC), a visualização dos próximos passos a serem seguidos torna-se cada vez mais simples, pois o diagrama de UC define cada traço e cada ligação entre os agentes do sistema (atores).

O modelo de Casos de Uso é um instrumento eficiente para determinação e documentação dos serviços a serem desempenhados pelo sistema. Ele é também um bom meio para comunicação com os clientes no processo de definição dos requisitos do sistema. (STADZISZ, 2002, p. 3)

Todas as interações e ligações e interações do sistema estão definidas no diagrama de UC. É através do diagrama de UC que são definidas as ações do sistema e quem vai interagir com o sistema por meio destas ações.

## 2. 2. DIAGRAMA DE CLASSES

Outra ferramenta que, elaborada após análise, auxilia no desenvolvimento do software, é denominada diagrama de classes.

Para a elaboração do diagrama de classes devem-se identificar as classes e seus atributos, identificar os relacionamentos entre eles e detalhar cada atributo para que assim haja maior precisão no levantamento dos dados e/ou correção de erros futuros.

No diagrama de classes serão especificados os atributos e os métodos de cada classe pertencente ao sistema.

Um diagrama de classes descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos estáticos existentes entre eles. Os diagramas de classe também mostram as propriedades e as operações de uma classe e as restrições que se aplicam à maneira como os objetos serão conectados. A UML utiliza a palavra característica como um termo geral que cobre as propriedades de uma classe. (FOWLER, 2005, p.52)

Através deste diagrama é possível visualizar as classes de forma geral, além de permitir com que seja especificada a maneira com que elas interagem entre si. Tal feito é fundamental para o desenvolvimento do sistema, tendo em vista que estas informações são primordiais em algumas etapas do desenvolvimento, como por exemplo, na criação do banco de dados.

### 2. 3. DIAGRAMA DE ATIVIDADES

Na definição feita por Martins (2007, p. 420), diagrama de atividades “é o tipo especial de máquina de estados que mostra o fluxo de atividades. Este diagrama é importante para modelar o comportamento dinâmico do sistema”.

Através do diagrama de atividades podem-se visualizar as respostas e os seguimentos que o sistema irá apresentar. Cada atividade possui um sequenciamento que é apresentado através deste modelo. É por meio deste que é possível tornar perceptíveis as ações que todas as atividades podem vir a desenvolver.

### 2. 4. MODELO ENTIDADE RELACIONAMENTO

O modelo de entidade relacionamento deve ser objetivo e direto para que não haja diversos pontos de vista do mesmo projeto. A função deste modelo é apresentar de forma gráfica os relacionamentos entre as entidades presentes no sistema.

Um modelo ER é um modelo formal, preciso, não ambíguo. Isto significa que diferentes leitores de um mesmo modelo ER devem sempre entender exatamente o mesmo. Tanto é assim, que um modelo ER pode ser usado como entrada de uma ferramenta CASE (Computer Aided Software Engineering) na geração de um banco de dados relacional. (HEUSER, 2009, p. 72)

Durante a elaboração do projeto, os envolvidos devem observar através do modelo entidade relacionamento e ter a mesma interpretação de como o sistema possuirá suas inter-relações.

As dependências entre as entidades devem ser apresentadas neste modelo para que haja um amplo entendimento de como os dados serão armazenados. Este modelo possui uma relação considerada direta com o banco de dados, no qual serão armazenados os dados registrados pelo sistema.

### **3. DESENVOLVIMENTO DO PROJETO**

Ao se desenvolver um projeto, o primeiro passo é definir quais serão os objetivos deste projeto, a quem ele deve atingir e por fim, as ferramentas que serão necessárias para a sua criação e desenvolvimento.

#### **3.1. OBJETIVO**

Este trabalho tem como objetivo o desenvolvimento de um software para controle e administração de uma academia personalizada. O sistema não tem como principal funcionalidade o controle de pagamentos das mensalidades dos seus clientes, mas sim se tornar uma ferramenta para auxiliar no acompanhamento do desempenho de cada aluno monitorado, armazenando os dados e medidas e apresentando resultados.

#### **3.2. PUBLICO-ALVO**

Este software visa atender academias que possuem um fluxo controlado de alunos, onde a dedicação ao desempenho de cada aluno é maior devido à baixa taxa de frequência do estabelecimento.

#### **3.3. JUSTIFICATIVA**

A preocupação com a saúde do corpo e da mente tem se tornado um fator relevante nos últimos anos. A busca por um corpo saudável e ideal tem se tornado uma das metas de grande parte da população. Levando em consideração o grande aumento na busca pelas academias supervisionadas, onde os treinos são monitorados e devidamente aplicados pelos profissionais da área, viu-se a necessidade de criação de um sistema que controle o desempenho dos clientes destas academias.

Tendo em vista que os públicos-alvo não são os grandes estabelecimentos e, conseqüentemente não exista filiais a fim de centralizarem informações, o software mais adequado é o desktop, pois não há a necessidade de compartilhamento de informações.

### 3.4. FERRAMENTAS UTILIZADAS

Desde a fase de desenvolvimento do projeto até a fase de implementação do sistema são necessárias ferramentas para colocar em prática o que foi planejado de forma conceitual.

O Astah é uma ferramenta que proporciona ao analista apresentar graficamente os modelos que envolvem o desenvolvimento do projeto. A utilização desta ferramenta é fundamental para aplicar os conceitos de modelagem utilizados no projeto. O software encontra-se disponível para download em [www.astah.net](http://www.astah.net).

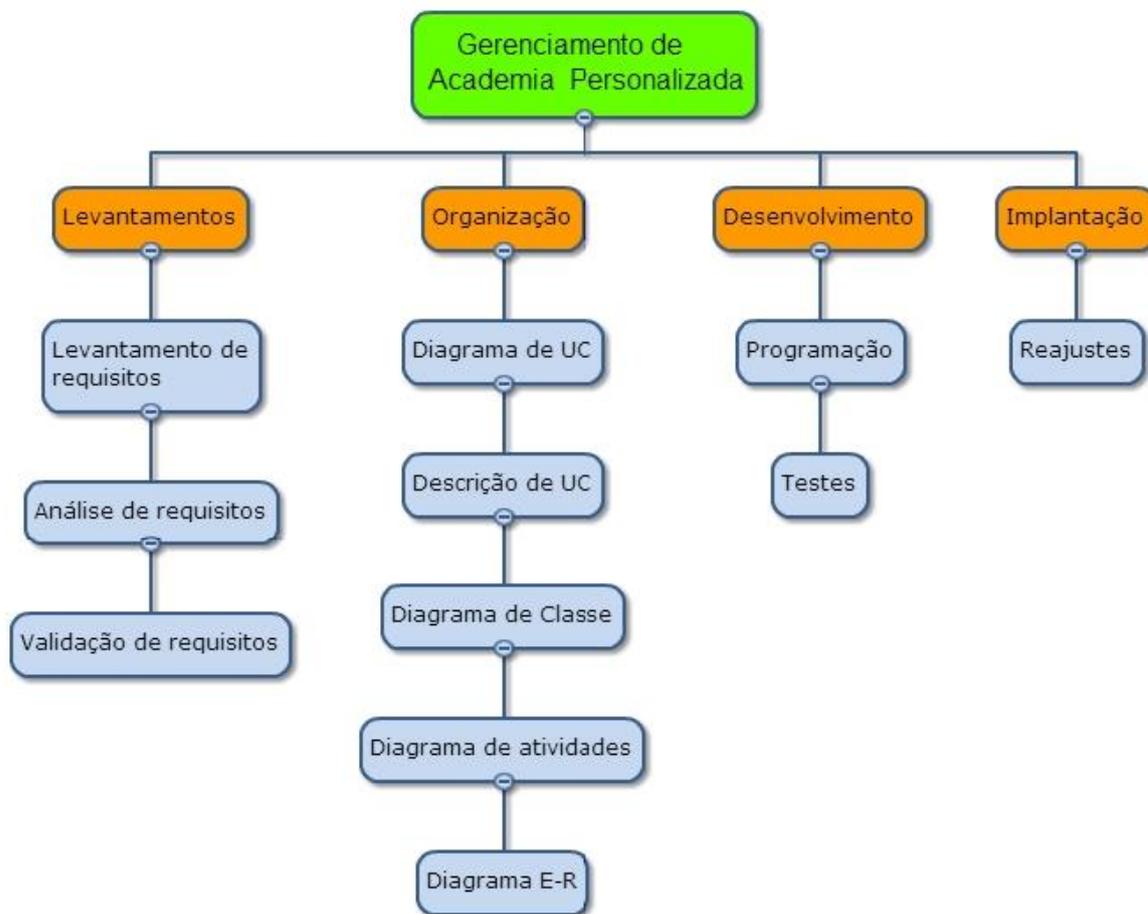
O trabalho foi desenvolvido através da linguagem C#, criação do grupo Microsoft, utilizando-se da orientação a objetos. O desenvolvimento foi realizado por meio do software de desenvolvimento Visual Studio, também de criação da Microsoft.

O armazenamento de dados também foi aplicado através das tecnologias da Microsoft, utilizando-se do Mysql, que possui como características portabilidade, maior velocidade no acesso às informações, praticidade, diversidade nos tipos de tabelas para armazenamento dos dados, dentre outras características mencionadas por MILANI (2007).

### 3.5. MÉTODO DE DESENVOLVIMENTO

Durante a fase de levantamentos de dados e possibilidades, o contato com clientes é fundamental para que o desenvolvimento do sistema alcance as metas almejadas e por fim garantir que o software desempenhe com precisão os papéis que lhe foram designados.

Inicialmente foi desenvolvida a Estrutura Analítica do Projeto (EAP), contemplando as principais entregas do projeto:



**Figura 1 - EAP (Estrutura Analítica do Projeto)**

### 3. 6. CASOS DE USO

Foram estabelecidos os principais eventos presentes no sistema para assim iniciar o planejamento dos casos de usos presentes no sistema.

A tabela 1 apresenta a lista de eventos existentes no sistema.

<b>Lista de Eventos</b>			
<b>Nº</b>	<b>Evento</b>	<b>Descrição</b>	<b>Ator</b>
1	Efetuar acesso	Efetuar o acesso ao Sistema	Usuário
2	Manter alunos	Efetuar Cadastro de alunos na academia	Usuário
3	Manter exercícios	Efetuar cadastro de exercícios	Usuário
4	Manter instrutores	Efetuar cadastro de instrutores habilitados no sistema	Usuário
5	Cadastrar plano de treino	Cadastrar planos de treinos de cada aluno	Instrutor
6	Cadastrar treinos	Efetuar cadastro de treinos	Instrutor
7	Manter usuários	Manter usuários no sistema	Usuário
8	Gerar desempenho	Gerar desempenho de alunos através de dados e medidas coletados de cada aluno	Instrutor
9	Registrar entrada/saída de aluno	Registrar a entrada e a saída dos alunos	Usuário/Aluno
10	Controlar pagamento	Controlar pagamento das mensalidades de cada aluno	Usuário
11	Emitir relatório de desempenho	Emitir relatório de desempenho de cada aluno	Instrutor
12	Emitir plano de treino	Emitir plano de treino para cada aluno	Instrutor
13	Validar acesso	Validar o acesso dos instrutores nas telas específicas	Instrutor
14	Gerar relatórios	Gerar relatórios de alunos, instrutores e usuários cadastrados	Usuário

**Tabela 1 - Lista de eventos**

A Figura 2 apresenta, de forma geral, o diagrama de casos de uso.

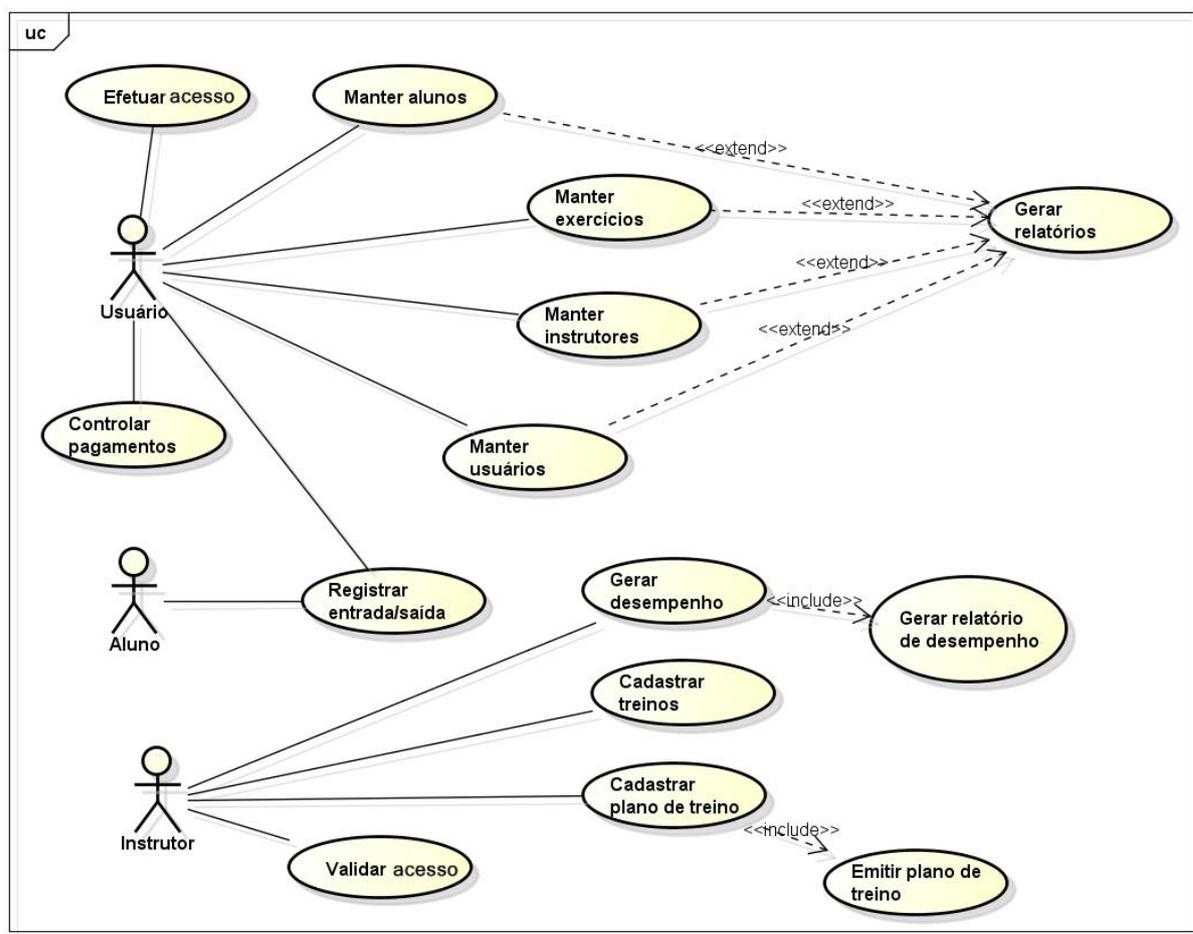


Figura 2 - Diagrama de UC Geral

### 3. 6. 1. Efetuar acesso

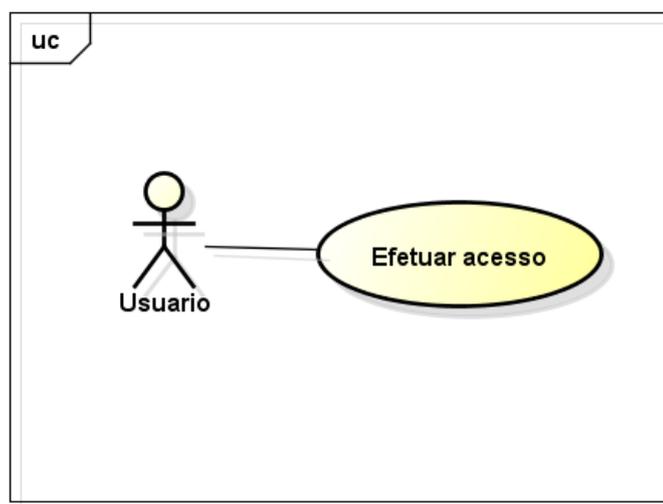


Figura 3 - CSU01 Efetuar acesso

<b>Nome da Use Case (CSU01)</b>	Efetuar acesso
<b>Ator (es)</b>	Usuário
<b>Pré-Condições</b>	Efetuar o acesso ao sistema
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa o sistema da academia.</li> <li>2. O sistema solicita a identificação.</li> <li>3. O usuário informa o código de usuário e senha.</li> <li>4. O sistema apresenta uma mensagem de boas vindas ao usuário no sistema.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. A senha informada pode estar incorreta. O sistema apresenta um alerta de erro na senha.</li> </ol>

Tabela 2 – Efetuar acesso

### 3.6.2. Manter alunos

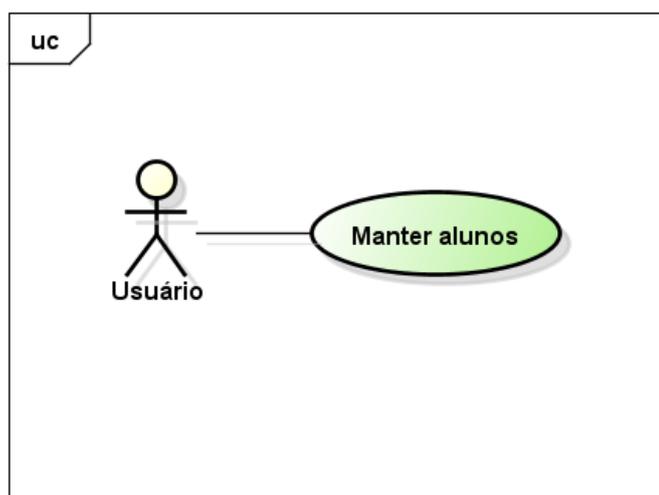


Figura 4 – CSU02 Manter alunos

<b>Nome da Use Case (CSU02)</b>	Manter alunos
<b>Ator (es)</b>	Usuário
<b>Pré-Condições</b>	Matricular e manter alunos no sistema
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O aluno informa o interesse em se matricular na academia.</li> <li>2. O usuário acessa o menu Cadastro de aluno.</li> <li>3. O aluno informa os dados pessoais que serão solicitados pelo sistema.</li> <li>4. O usuário confirma os dados.</li> <li>5. O sistema emite a mensagem “Aluno cadastrado com sucesso”.</li> <li>6. O sistema cadastra o aluno.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O usuário poderá alterar os dados dos alunos já cadastrados</li> <li>2. O usuário poderá excluir o aluno do sistema.</li> </ol>

Tabela 3 - Manter alunos

### 3. 6. 3. Manter exercícios

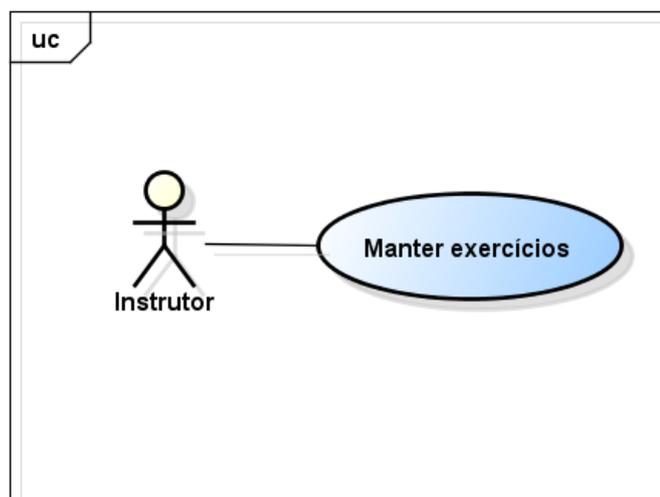
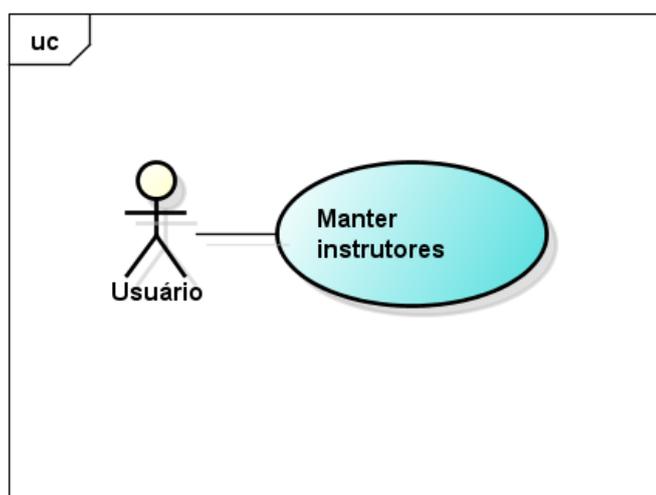


Figura 5 – CSU03 Manter exercícios

<b>Nome da Use Case (CSU03)</b>	Manter exercícios
<b>Ator (es)</b>	Instrutor
<b>Pré-Condições</b>	Cadastrar um novo tipo de exercício
<b>Cenário principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa o menu cadastro de exercício.</li> <li>2. O usuário informa os dados solicitados pelo sistema sobre o novo exercício a ser cadastrado.</li> <li>3. Os dados são confirmados pelo usuário.</li> <li>4. O sistema cadastra o exercício.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O instrutor pode cancelar a operação a qualquer momento ou não confirmar o lançamento.</li> <li>2. Um exercício pode excluído do sistema ou</li> </ol>

**Tabela 4 - Manter exercícios**

### 3. 6. 4. Manter instrutores

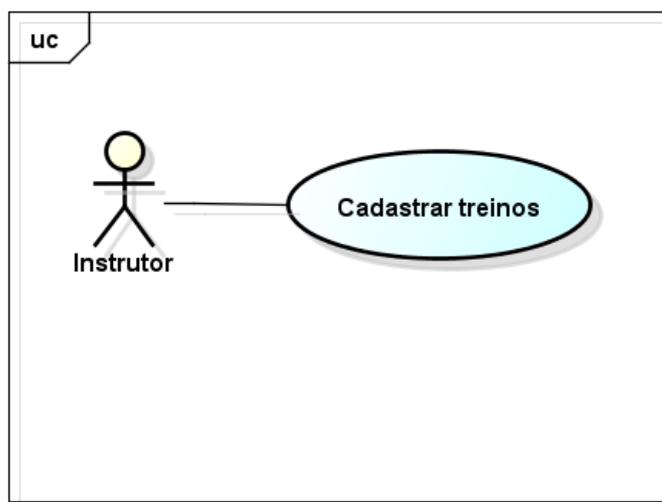


**Figura 6 – CSU04 Manter instrutores**

<b>Nome da Use Case (CSU04)</b>	Manter instrutores
<b>Ator (es)</b>	Usuário
<b>Pré-Condições</b>	Manter instrutores no sistema
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa o menu de cadastro de instrutores.</li> <li>2. O instrutor informa os dados pessoais que serão solicitados pelo sistema.</li> <li>3. O usuário confirma os dados.</li> <li>4. O sistema emite a mensagem “Instrutor cadastrado com sucesso”.</li> <li>5. O sistema cadastra o instrutor.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O usuário poderá alterar os dados posteriormente.</li> <li>2. O instrutor poderá ser excluído do sistema.</li> </ol>

**Tabela 5 - Manter instrutores**

### 3. 6. 5. Cadastrar treinos



**Figura 7– CSU05 Manter treino**

<b>Nome da Use Case (CSU05)</b>	Cadastrar treinos
<b>Ator (es)</b>	Instrutor
<b>Pré-Condições</b>	Cadastrar treinos no sistema
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema solicita a identificação do instrutor na tela de cadastro de treinos.</li> <li>2. O instrutor digita seus dados e senha</li> <li>3. O sistema confirma a identificação.</li> <li>4. O sistema abre a tela de cadastro de treino.</li> <li>5. O instrutor informa a matrícula do aluno e o tipo de treino que irá cadastrar.</li> <li>6. O sistema armazena o cadastro para ser adicionado posteriormente a um plano de treino.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O sistema pode não confirmar a identificação.</li> </ol>

Tabela 6 – Cadastrar treinos

### 3. 6. 6. Cadastrar plano de treino

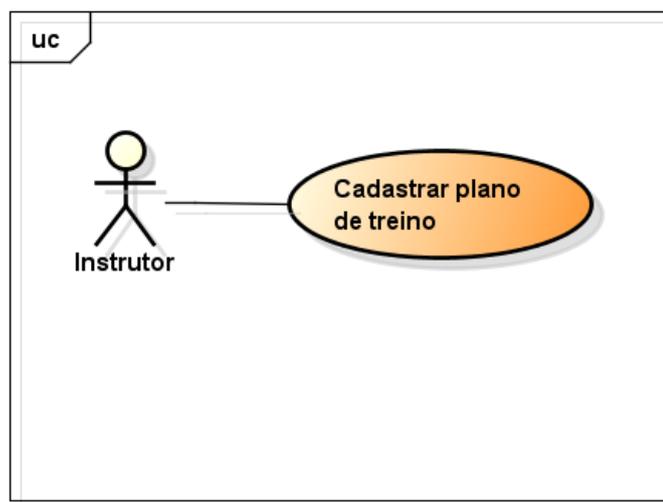
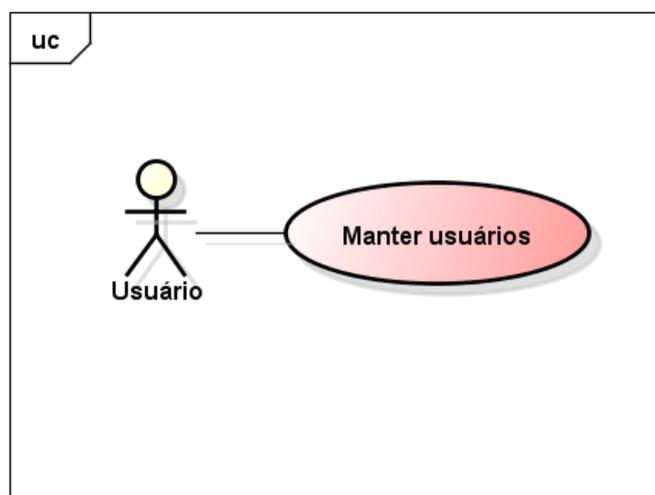


Figura 8 – CSU06 Cadastrar plano de treino

<b>Nome da Use Case (CSU06)</b>	Cadastrar planos de treino
<b>Ator (es)</b>	Instrutor
<b>Pré-Condições</b>	Cadastrar planos de treinos no sistema
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema solicita que seja efetuada a identificação do instrutor na tela de cadastro de planos de treino.</li> <li>2. O instrutor digita o código de acesso e senha para se autenticar no sistema.</li> <li>3. O sistema abre a tela de desempenhos.</li> <li>4. O instrutor informa o treino já cadastrado que deverá ser adicionado ao plano.</li> <li>5. O instrutor informa as repetições e a carga de cada exercício.</li> <li>6. O sistema armazena o plano.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O sistema pode não confirmar a identificação.</li> </ol>

**Tabela 7 - Cadastrar plano de treino**

### 3. 6. 7. Manter usuários

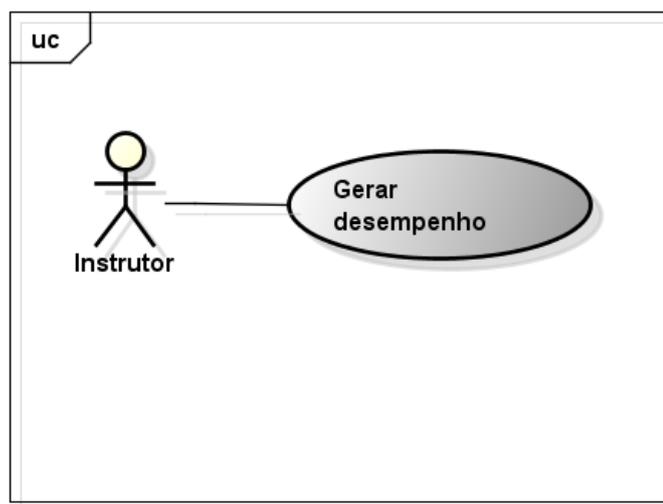


**Figura 9 – CSU07 Manter usuários**

<b>Nome da Use Case (CSU07)</b>	Manter usuários
<b>Ator (es)</b>	Usuário
<b>Pré-Condições</b>	Cadastrar e manter usuários no sistema
<b>Cenário principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa o menu de cadastro de usuários.</li> <li>2. O sistema solicita os dados que deverão ser preenchidos pelo usuário.</li> <li>3. O sistema gera o código do usuário e o cadastra no sistema.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1.O usuário poderá alterar os dados dos usuários.</li> <li>2.O usuário poderá excluir um usuário quando necessário, após confirmação.</li> </ol>

**Tabela 8 – Manter usuários**

### 3. 6. 8. Gerar desempenho

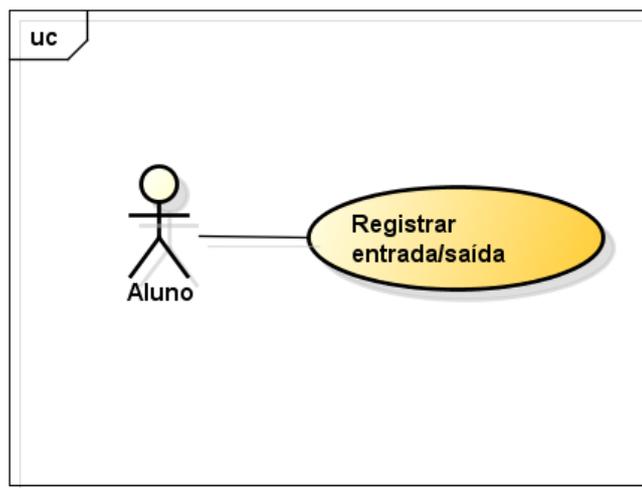


**Figura 10– CSU08 Gerar desempenho**

<b>Nome da Use Case (CSU08)</b>	Gerar desempenho
<b>Ator (es)</b>	Instrutor
<b>Pré-Condições</b>	Gerar desempenho de um aluno já cadastrado
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O instrutor deverá se identificar.</li> <li>2. O sistema confirma o instrutor no sistema.</li> <li>3. O instrutor informa o aluno o qual irá gerar o desempenho e preenche os campos solicitados.</li> <li>4. O sistema calcula com base nos dados e cadastra.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. A senha pode estar incorreta.</li> <li>2. Os dados podem não serem aceitos.</li> </ol>

**Tabela 9 – Gerar desempenho**

### 3. 6. 9. Registrar entrada/saída

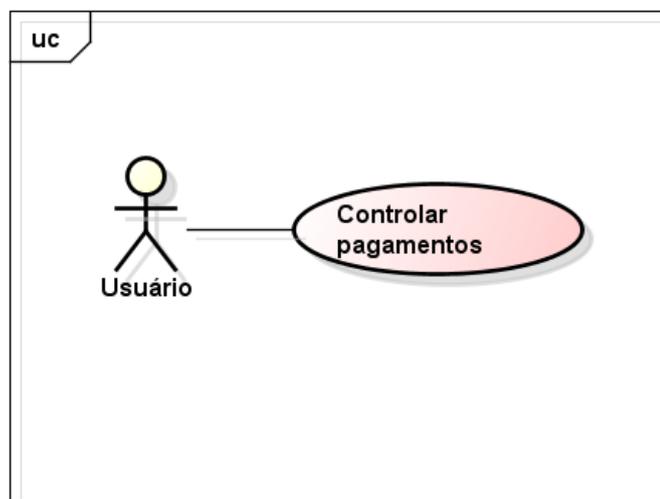


**Figura 11– CSU09 Registrar entrada/saída**

<b>Nome da Use Case (CSU09)</b>	Registrar entrada/saída
<b>Ator (es)</b>	Aluno
<b>Pré-Condições</b>	Registrar a entrada e saída de um aluno
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Na tela de registro de entrada e saída, o usuário informa a matrícula do aluno.</li> <li>2. O usuário seleciona a opção entrada ou saída.</li> <li>3. O aluno digita a senha.</li> <li>4. O sistema registra a operação.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O aluno pode informar a senha incorreta.</li> </ol>

**Tabela 10 - Registrar entrada/saída**

### 3. 6. 10. Controlar pagamentos

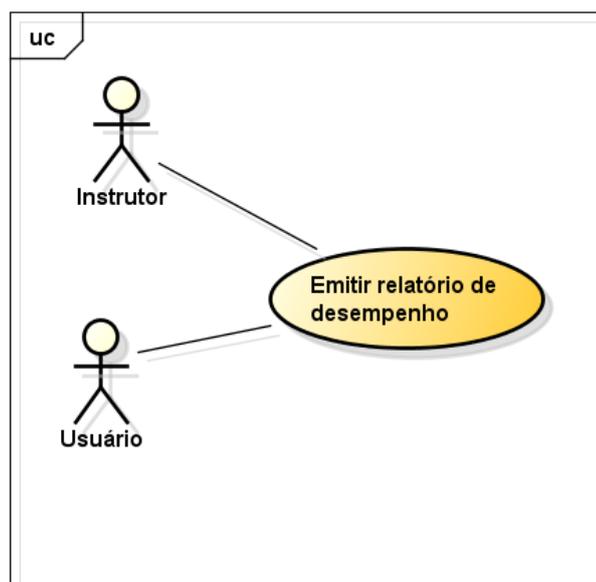


**Figura 12 – CSU10 Controlar pagamentos**

<b>Nome da Use Case (CSU10)</b>	Controlar pagamentos
<b>Ator (es)</b>	Usuário
<b>Pré-Condições</b>	Controlar o pagamento das mensalidades
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário consulta o aluno o qual deseja pagar a mensalidade.</li> <li>2. O sistema apresenta os pagamentos O usuário informa a mensalidade desejada.</li> <li>3. O sistema confirma o pagamento.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O aluno pode ter mais de um pagamento pendente e/ou em atraso.</li> </ol>

**Tabela 11 - Controlar pagamentos**

### 3. 6. 11. Emitir relatório de desempenho

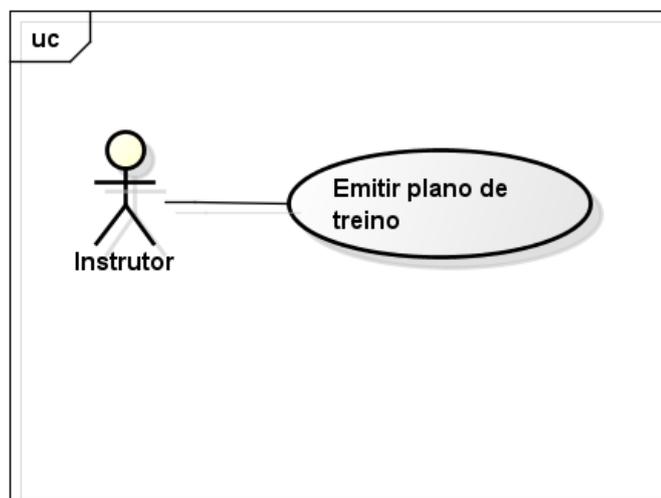


**Figura 13 – CSU11 Emitir relatório de desempenho**

<b>Nome da Use Case (CSU11)</b>	Emitir relatório de desempenho
<b>Ator (es)</b>	Usuário/Instrutor
<b>Pré-Condições</b>	Emitir relatório, já cadastrado, que apresenta o desempenho do aluno
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Na tela emissão de relatório de desempenho, deve ser informado o código do aluno.</li> <li>2. O sistema apresenta os desempenhos cadastrados para este aluno.</li> <li>3. O usuário/instrutor seleciona o desempenho desejado.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O sistema pode não encontrar registros.</li> </ol>

**Tabela 12 - Emitir relatório de desempenho**

### 3. 6. 12. Emitir plano de treino

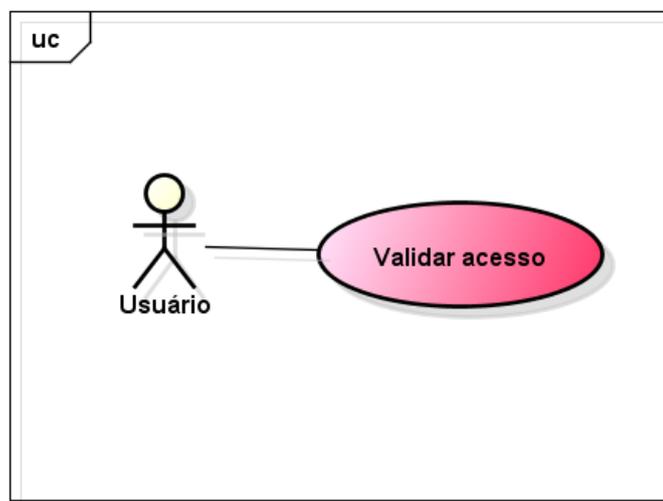


**Figura 14 – CSU12 Emitir plano de treino**

<b>Nome da Use Case (CSU12)</b>	Emitir plano de treino
<b>Ator (es)</b>	Instrutor
<b>Pré-Condições</b>	Emitir o plano de treino já cadastrado
<b>Cenário Principal</b>	<ol style="list-style-type: none"><li>1. O instrutor seleciona o plano de treino que deseja utilizar.</li><li>2. O sistema emite o plano para que o aluno possa acompanhá-lo.</li></ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"><li>1. O sistema pode não encontrar registros.</li></ol>

**Tabela 13 – Emitir plano de treino**

### 3. 6. 13. Validar acesso

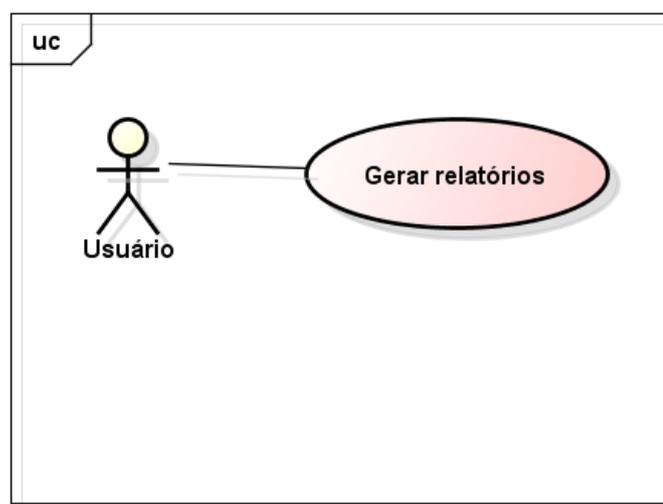


**Figura 15 – CSU13 Validar acesso**

<b>Nome da Use Case (CSU13)</b>	Validar acesso
<b>Ator (es)</b>	Instrutor
<b>Pré-Condições</b>	Validar o acesso de instrutores nas telas específicas de instrutores
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema solicita que seja informado o código e a senha do instrutor para prosseguir com o acesso.</li> <li>2. O instrutor informa os dados e confirma a operação.</li> <li>3. O sistema valida os dados e libera o acesso.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. O sistema pode não identificar o instrutor devido à senha errada e/ou código inválido.</li> <li>2. O sistema notifica na tela caso haja erro de senha e/ou código inválido.</li> </ol>

**Tabela 14 - Validar acesso**

### 3. 6. 14. Gerar relatórios



**Figura 16 – CSU14 Gerar relatórios**

<b>Nome da Use Case (CSU14)</b>	Gerar relatórios
<b>Ator (es)</b>	Usuário
<b>Pré-Condições</b>	Gerar relatórios de alunos, instrutores e usuários cadastrados.
<b>Cenário Principal</b>	<ol style="list-style-type: none"><li>1. No menu de acesso a relatórios, o usuário deverá informar qual o tipo de relatório deseja executar.</li><li>2. O sistema gera o relatório e o apresenta em tela.</li></ol>
<b>Cenário alternativo</b>	<ol style="list-style-type: none"><li>1. O sistema pode não encontrar registros para os parâmetros solicitados.</li></ol>

**Tabela 15 – Gerar relatórios**

### 3. 7. DIAGRAMA DE CLASSES

A figura abaixo apresenta as classes que foram criadas para contemplar o desenvolvimento do sistema.

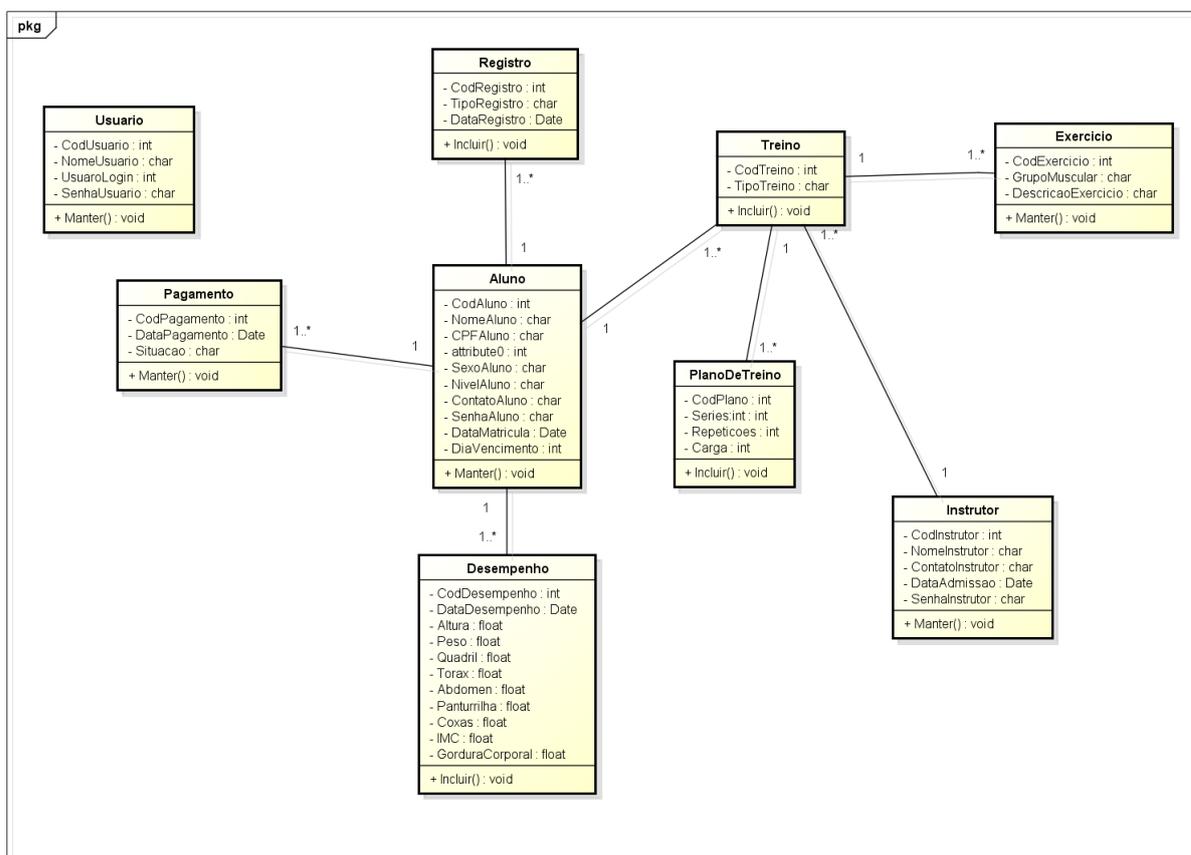


Figura 17 – Diagrama de classes

### 3. 8. DIAGRAMAS DE ATIVIDADES

Os tópicos a seguir apresentam as figuras que representam os diagramas de atividades desenvolvidas pelo sistema.

### 3. 8. 1. Atividade Efetuar Acesso

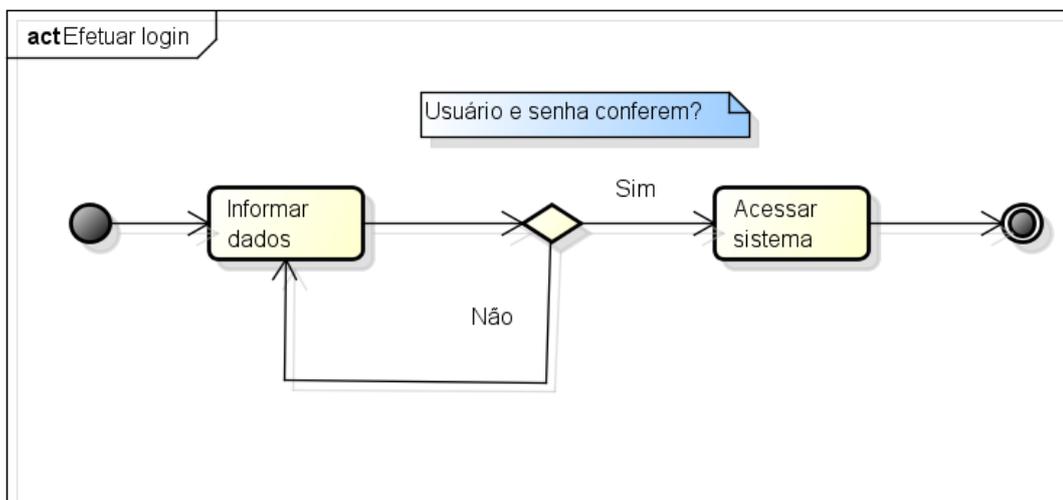


Figura 18 – Atividade Efetuar acesso

### 3. 8. 2. Atividade Manter alunos

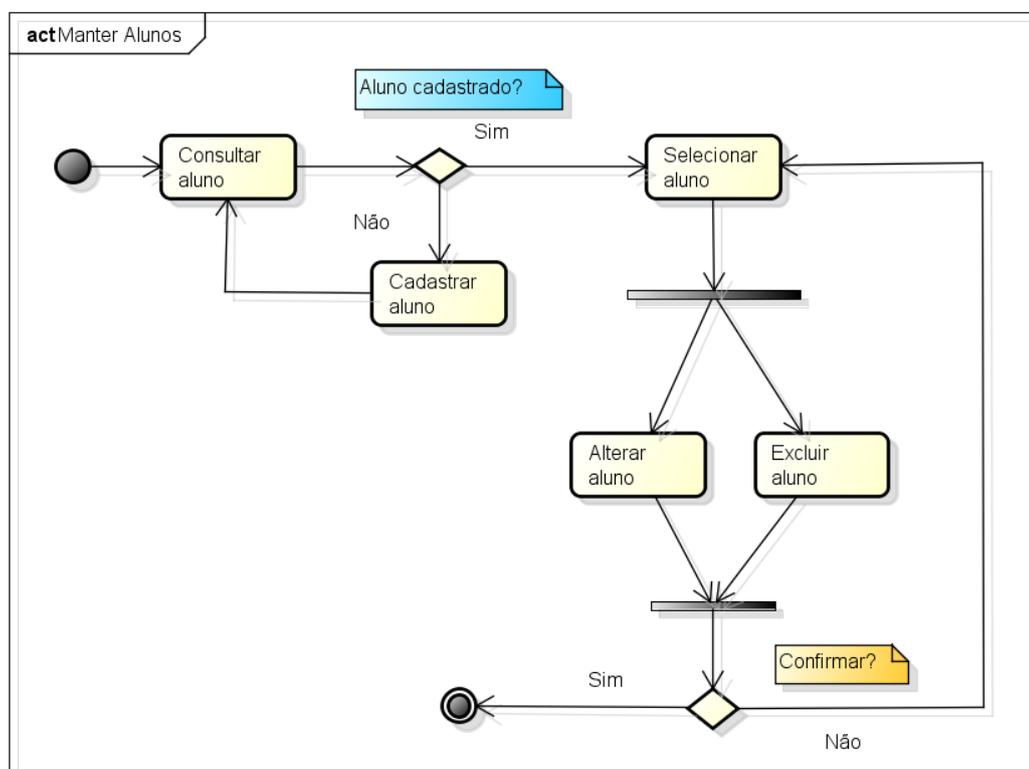


Figura 19 – Atividade Manter alunos

### 3. 8. 3. Atividade Manter exercícios

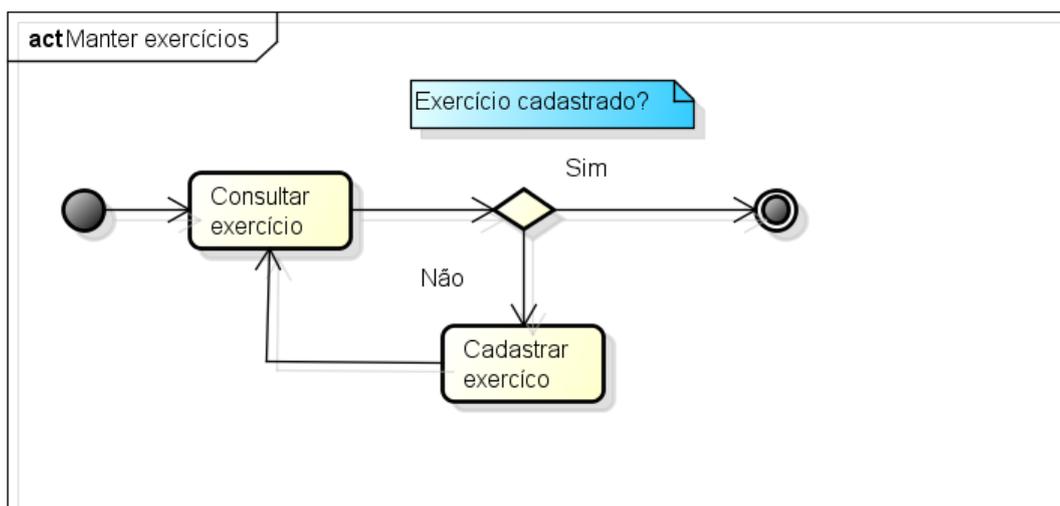


Figura 20 – Atividade Manter exercícios

### 3. 8. 4. Atividade Manter instrutores

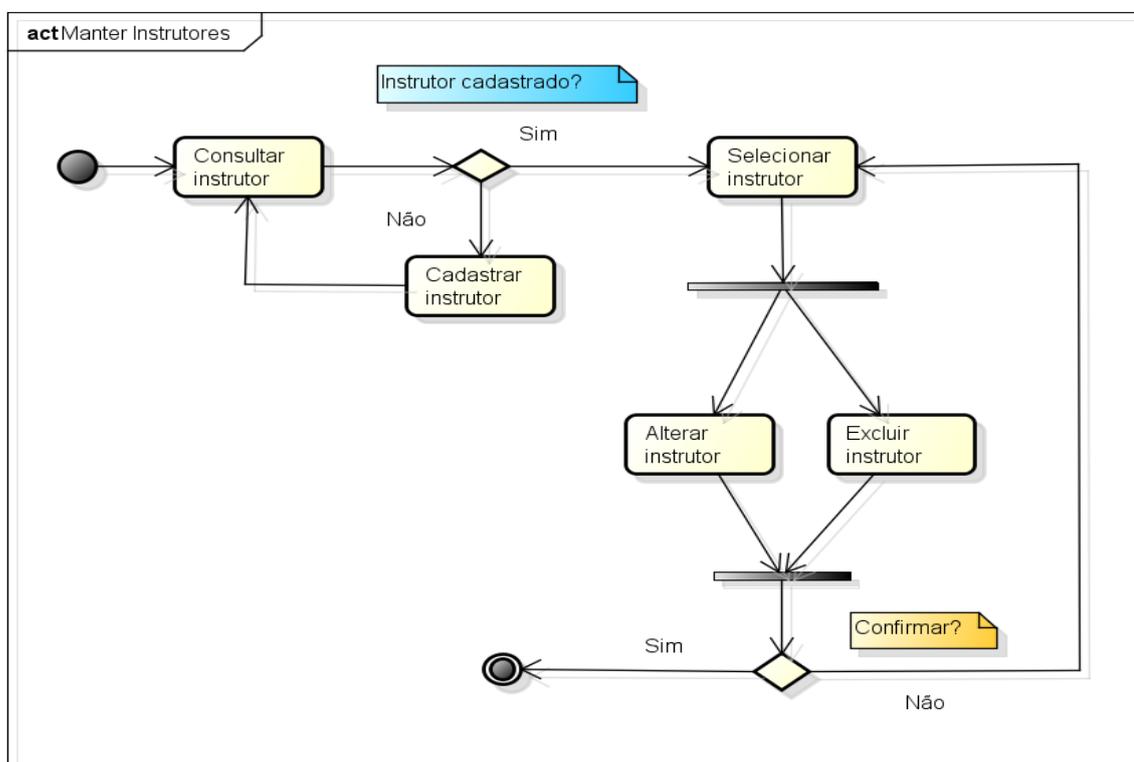


Figura 21 – Atividade Manter instrutores

### 3. 8. 5. Atividade Cadastrar treinos

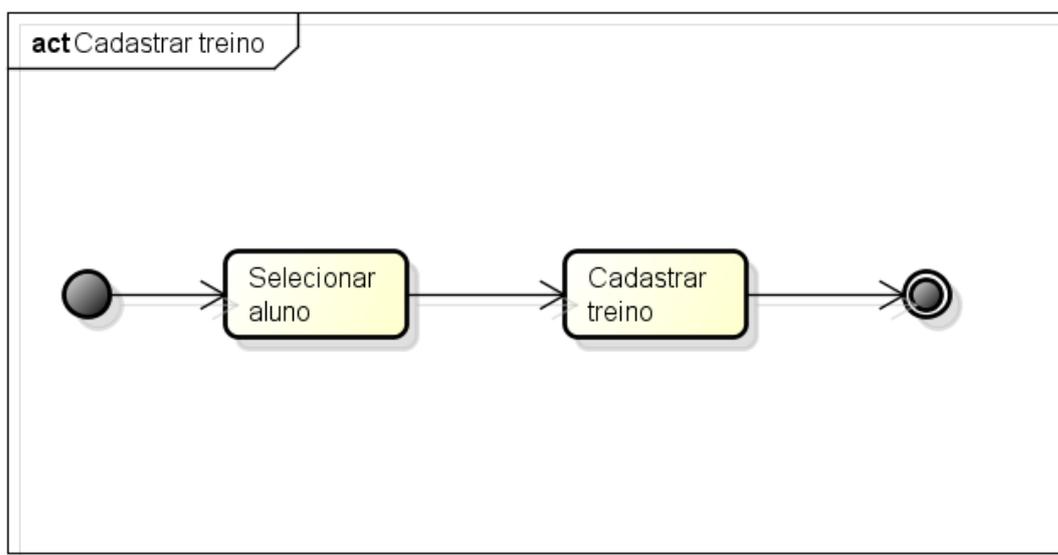


Figura 22 - Atividades Cadastrar treinos

### 3. 8. 6. Atividade Cadastrar planos de treino

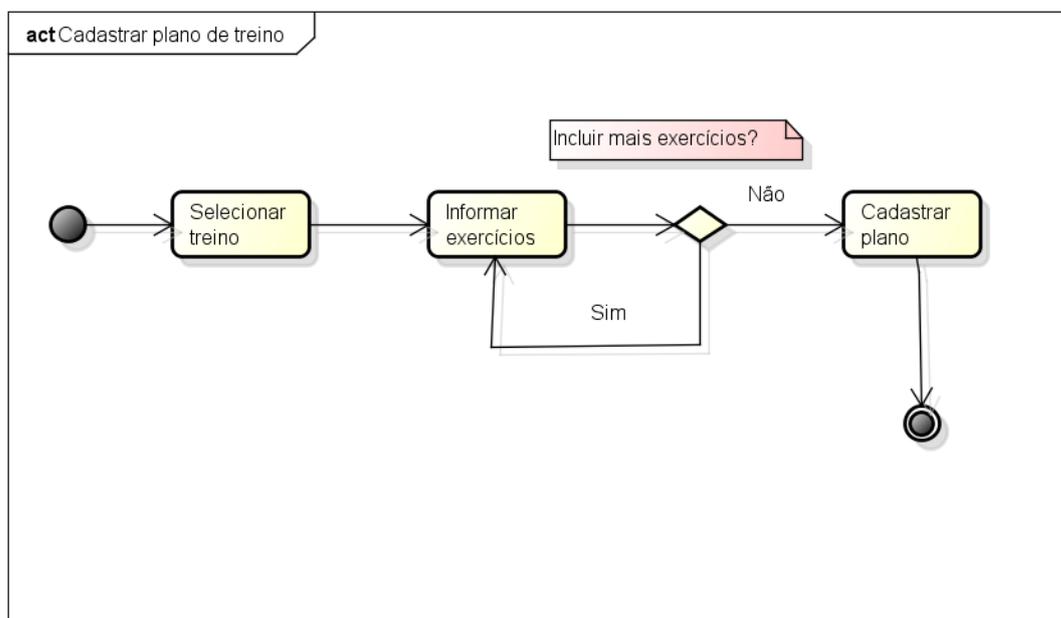


Figura 23 – Atividade Cadastrar planos de treino

### 3. 8. 7. Atividade Manter usuários

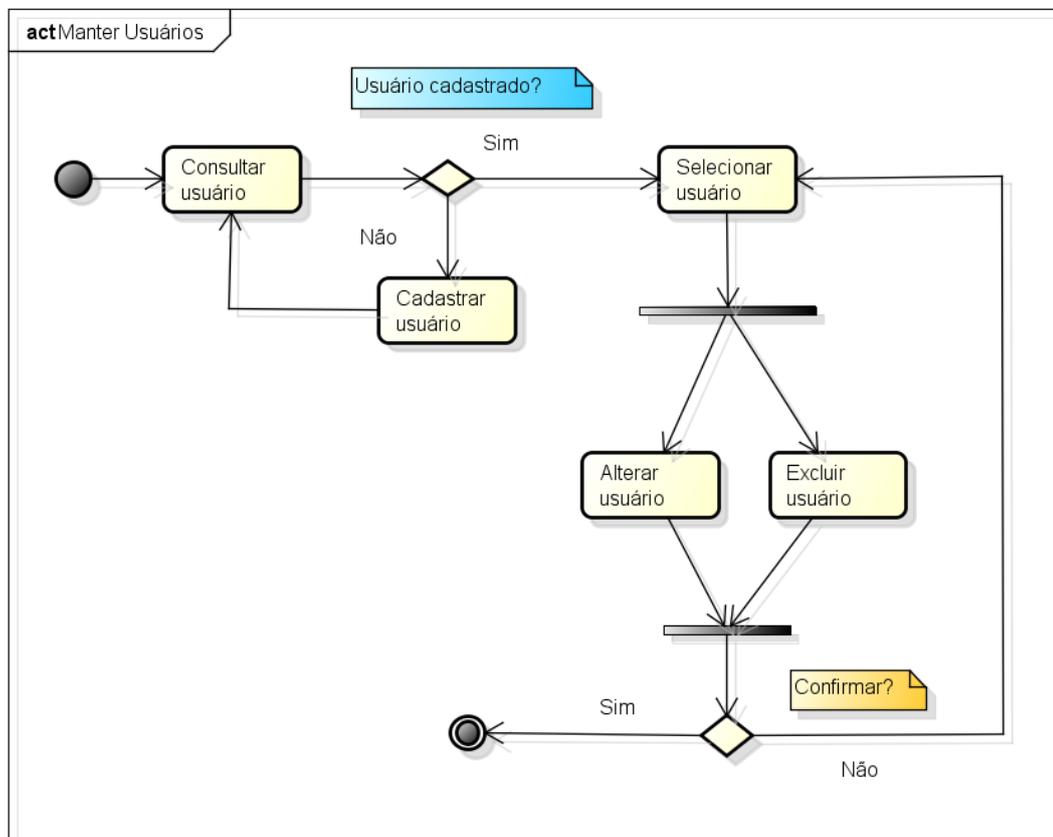


Figura 24 – Atividade Manter usuários

### 3. 8. 8. Atividade Cadastrar desempenhos

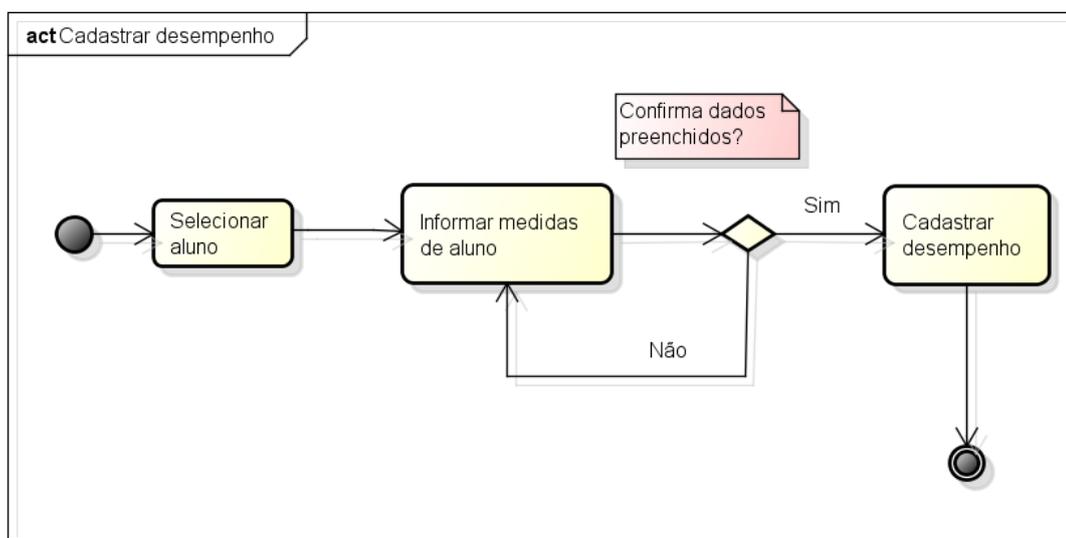


Figura 25 – Atividade Cadastrar desempenhos

### 3. 8. 9. Atividade Registrar Entrada/Saída

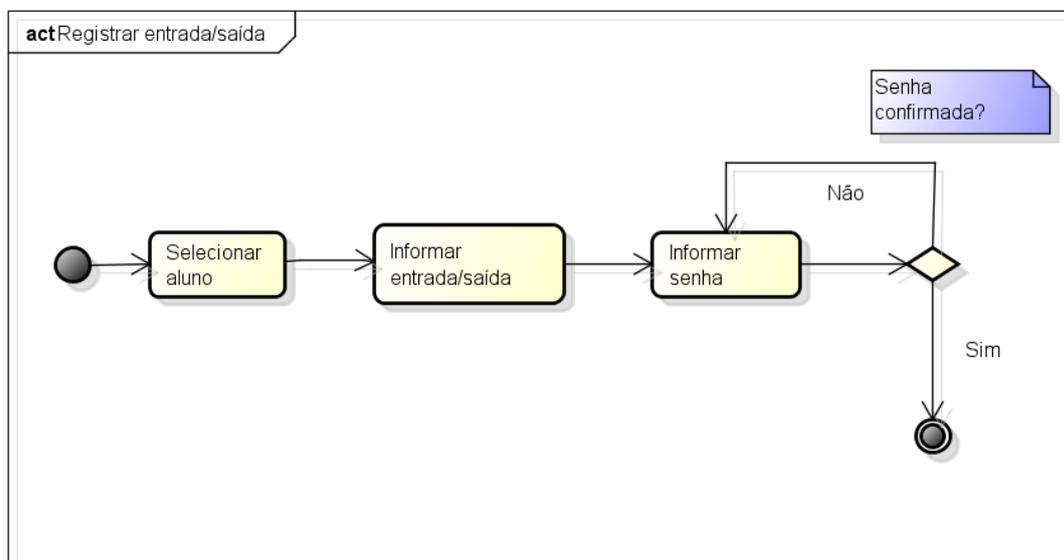


Figura 26 – Atividade Registrar Entrada/Saída

### 3. 8. 10. Atividade Controlar pagamentos

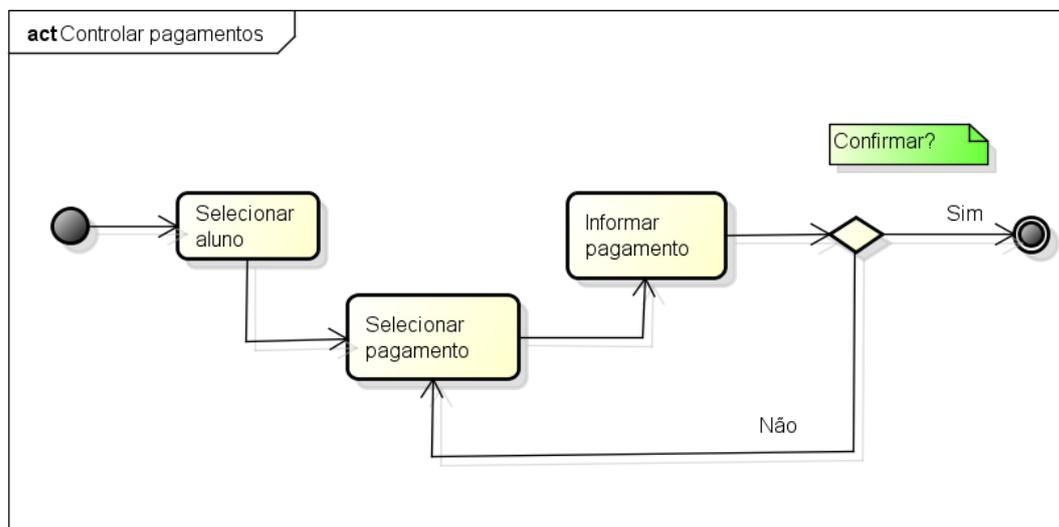


Figura 27 – Atividade Controlar pagamentos

### 3. 8. 11. Atividade Gerar relatório de desempenho

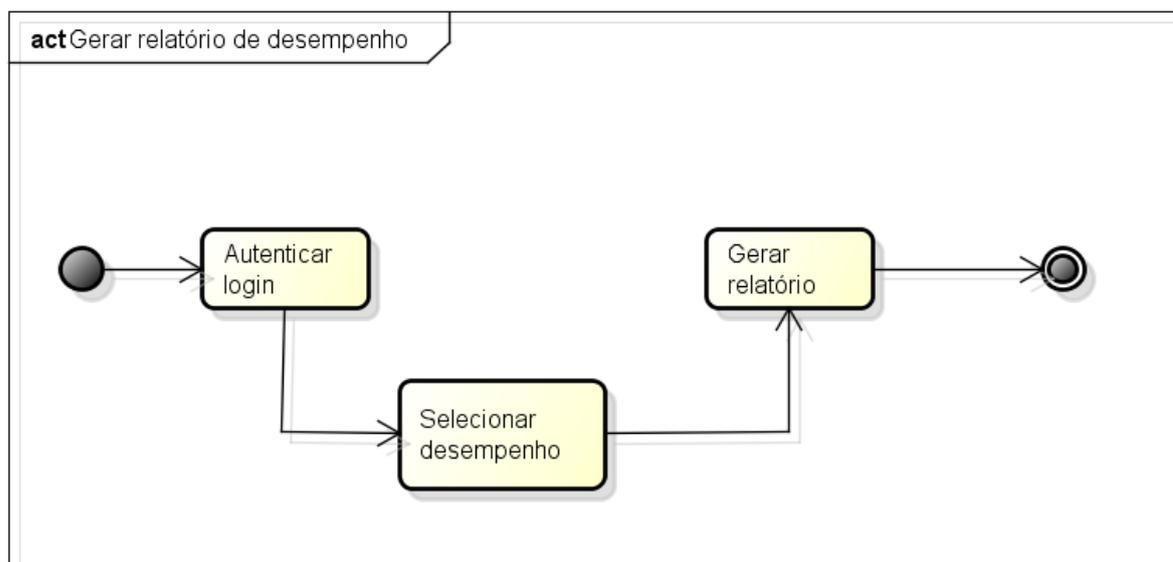


Figura 28 – Atividade Gerar relatório de desempenho

### 3. 8. 12. Atividade Gerar plano de treino

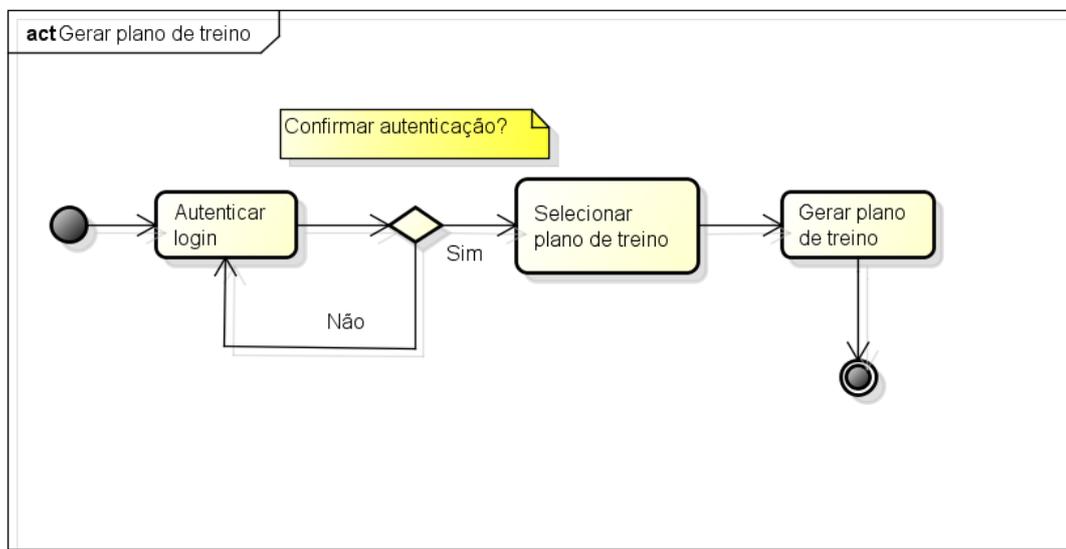


Figura 29 – Atividade Gerar plano de treino

### 3. 8. 13. Atividade Validar acesso

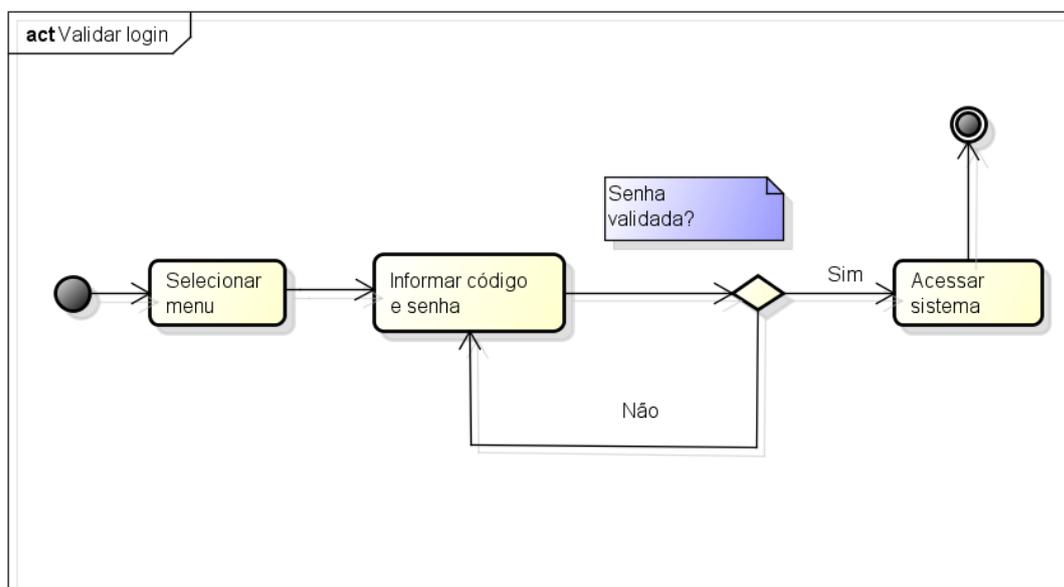


Figura 30 – Atividade Validar acesso

### 3. 8. 14. Atividade Gerar relatórios

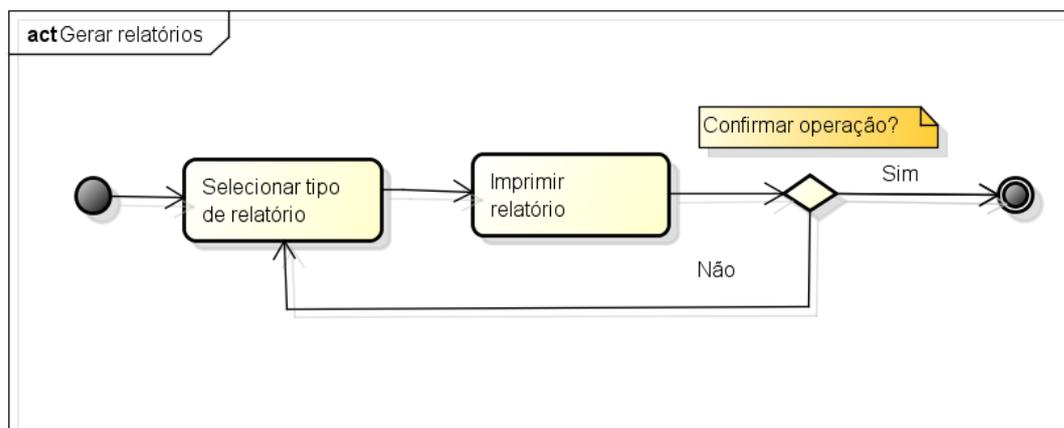
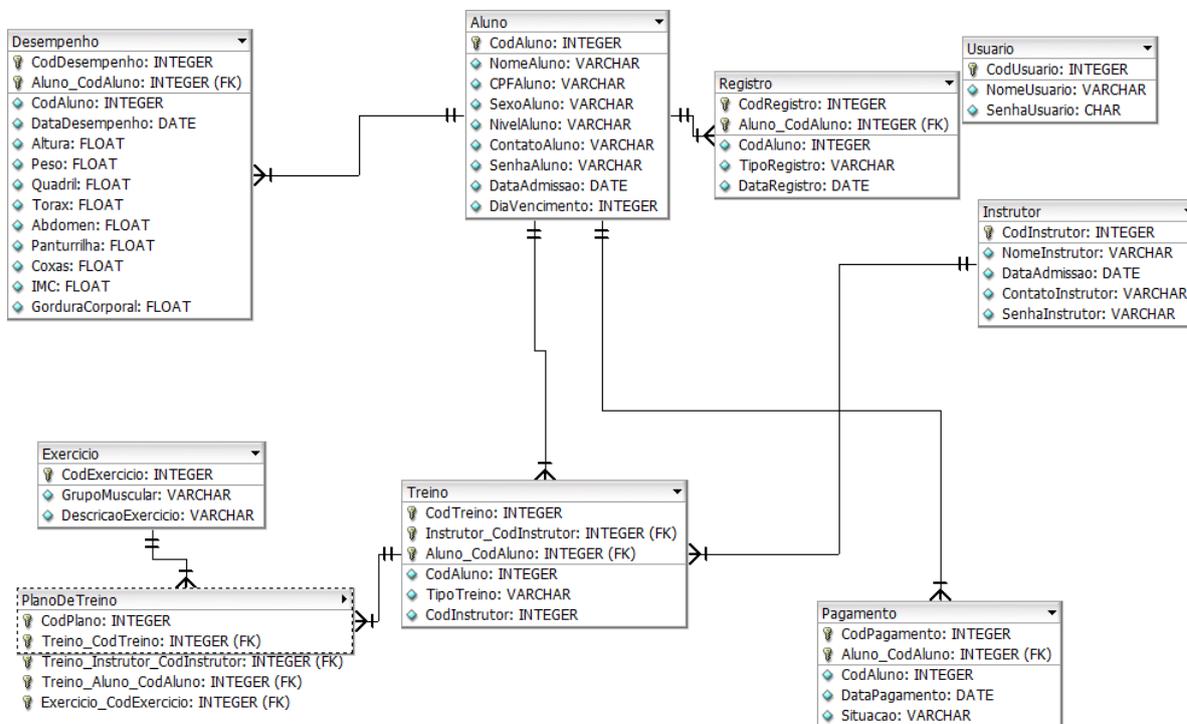


Figura 31 – Atividade Gerar relatórios

### 3. 9. MODELO ENTIDADE-RELACIONAMENTO

O modelo ER foi desenvolvido para apresentar as dependências entre as classes e a forma como elas se relacionam entre si.



**Figura 32 – Modelo Entidade-Relacionamento**

## 4. IMPLEMENTAÇÃO DO SISTEMA

O sistema foi planejado visando o desenvolvimento em camadas. O desenvolvimento em camadas tem como funcionalidade a separação e a organização na programação, pois os códigos são divididos em camadas que, logicamente possuem responsabilidades divergentes, porém possuem uma comunicação entre elas. (CALÇADO, 2005, p.34-35)

Divide-se a programação do sistema em três camadas: camada de modelo, onde são implementadas as classes, camada de persistência, nela há a interação direta com o banco de dados e a camada de negócio armazena o plano de negócio do sistema. Grande parte da complexidade do programa é apresentada e resolvida na camada de negócio.

CALÇADO (2005) apresenta como vantagens do desenvolvimento em camadas a separação do código; a independência das camadas – onde uma parte do sistema pode ser alterada sem prejudicar outra; praticidade em corrigir erros; além de auxiliar no entendimento dos códigos, facilitando a comunicação entre os programadores. O autor considera como desvantagens as limitações das tecnologias, a complexidade de programação e a complicação de sistemas simples, que, muitas vezes, não são viáveis de implementação das camadas.

### 4.1. CAMADA MODEL

A camada Model, como já mencionado anteriormente, apresenta as classes existentes no sistema.

A figura 33 representa a classe Aluno presente no projeto:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace GerenciamentoAcademia.Model
{
    public class Aluno
    {
        public int CodAluno { get; set; }
        public string NomeAluno { get; set; }
        public int CPFAluno { get; set; }
        public string SexoAluno { get; set; }
        public string NivelAluno { get; set; }
        public string ContatoAluno { get; set; }
        public string SenhaAluno { get; set; }
        public string ConfirmaSenhaAluno { get; set; }
        public DateTime DataMatricula { get; set; }
        public int DiaVencimento { get; set; }
    }
}

```

**Figura 33 – Classe aluno model**

## 4.2. CAMADA DAL

A camada DAL, camada de acesso a dados, estabelece a comunicação com o banco de dados, implementando assim a conexão direta dos dados. Nela estão presentes as funções (void) de movimentação de dados (incluir, alterar e excluir).

As figuras 34, 35 e 36 apresentam as funções incluir, alterar e excluir, respectivamente.

```

public void Incluir(Model.Instrutor instrutor)
{
    //conexão
    SqlConnection cn = new SqlConnection();
    try
    {
        cn.ConnectionString = Dados.StringDeConexao;
        //command
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = cn;
        cmd.CommandText = "insert into Instrutores (NomeInstrutor,ContatoInstrutor,DataAdmissao) values (@NomeInstrutor, @ContatoInstrutor, @DataAdmissao);select @@IDENTITY";
        cmd.Parameters.AddWithValue("@NomeInstrutor", instrutor.NomeInstrutor);
        cmd.Parameters.AddWithValue("@ContatoInstrutor", instrutor.ContatoInstrutor);
        cmd.Parameters.AddWithValue("@DataAdmissao", instrutor.DataAdmissao);

        cn.Open();
        instrutor.CodInstrutor = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
    finally
    {
        cn.Close();
    }
}

```

**Figura 34 – Void incluir**

```

public void Alterar(Model.Usuario usuario)
{
    //conexao
    SqlConnection cn = new SqlConnection();
    try
    {
        cn.ConnectionString = Dados.StringDeConexao;
        //command
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = cn;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "update Usuarios set NomeUsuario=@NomeUsuario, SenhaUsuario=@SenhaUsuario, ConfirmaSenhaUsuario=@ConfirmaSenhaUsuario where CodUsuario=@CodUsuario;";
        cmd.Parameters.AddWithValue("@CodUsuario", usuario.CodUsuario);
        cmd.Parameters.AddWithValue("@NomeUsuario", usuario.NomeUsuario);
        cmd.Parameters.AddWithValue("@SenhaUsuario", usuario.SenhaUsuario);
        cmd.Parameters.AddWithValue("@ConfirmaSenhaUsuario", usuario.ConfirmaSenhaUsuario);

        cn.Open();
        cmd.ExecuteNonQuery();
    }
    finally
    {
        cn.Close();
    }
}

```

**Figura 35 – Void alterar**

```

public void Excluir(int cod)
{
    SqlConnection cn = new SqlConnection();
    try
    {
        cn.ConnectionString = Dados.StringDeConexao;
        //command
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = cn;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "delete from Alunos where CodAluno=" + cod + ";";

        cn.Open();
        cmd.ExecuteNonQuery();
    }
    finally
    {
        cn.Close();
    }
}

```

**Figura 36 – Void excluir**

### 4. 3. CAMADA BLL

A camada BLL, camada de lógica de negócio, ainda se comunica diretamente com a camada de persistência, DAL, instanciando objetos.

Na figura 37 é possível visualizar os objetos sendo instanciados através da interação entre as duas camadas.

```
public class UsuarioBLL
{
    private UsuarioDAL objDAL;

    public UsuarioBLL()
    {
        objDAL = new UsuarioDAL();
    }

    public void Incluir(Model.Usuario usuarios)
    {
        DAL.UsuarioDAL obj = new DAL.UsuarioDAL();
        obj.Incluir(usuarios);
    }

    public void Excluir(int cod)
    {
        DAL.UsuarioDAL obj = new DAL.UsuarioDAL();
        obj.Excluir(cod);
    }

    public void Alterar(Model.Usuario usuarios)
    {
        DAL.UsuarioDAL obj = new DAL.UsuarioDAL();
        obj.Alterar(usuarios);
    }
}
```

**Figura 37 – Camada BLL**

#### 4.4. CAMADA GRÁFICA

A apresentação gráfica proporciona ao usuário final uma interação com o sistema de forma com que não haja divergências ou má interpretação de funcionalidades.

Através das telas desenhadas pelo software de desenvolvimento, o usuário poderá inserir os dados e administrá-lo da forma adequada.

As figuras a seguir apresentam algumas telas do sistema, as quais proporcionam ao usuário final uma interação direta com o software.

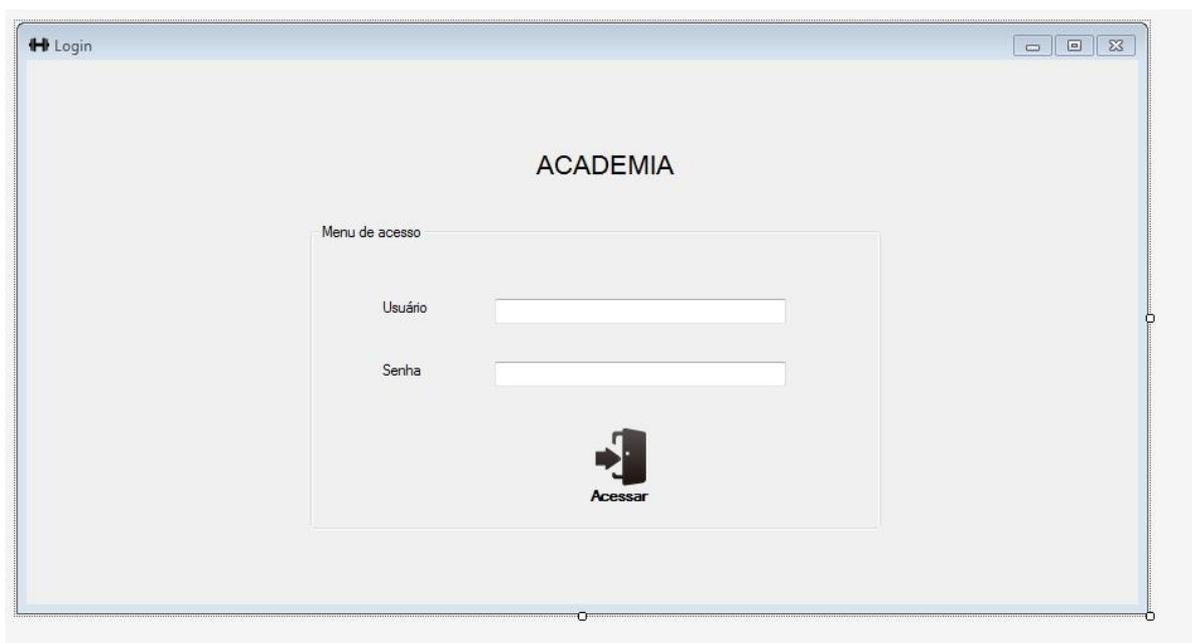


Figura 38 – Tela de acesso

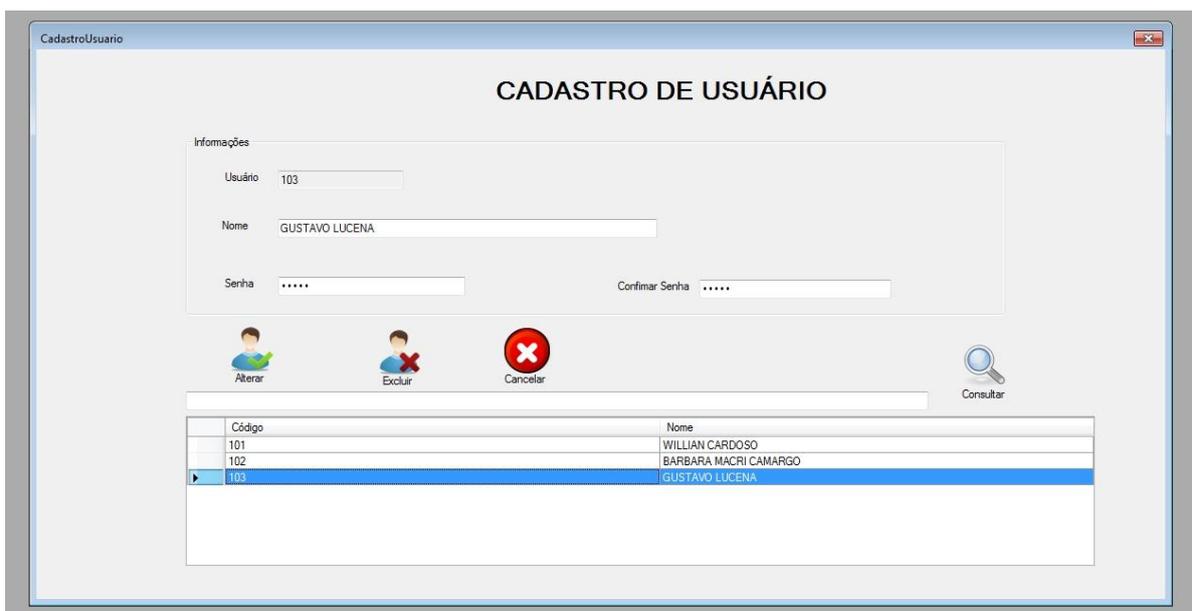


Figura 39– Tela de Cadastro de usuário



Figura 40 – Tela de verificação de acesso

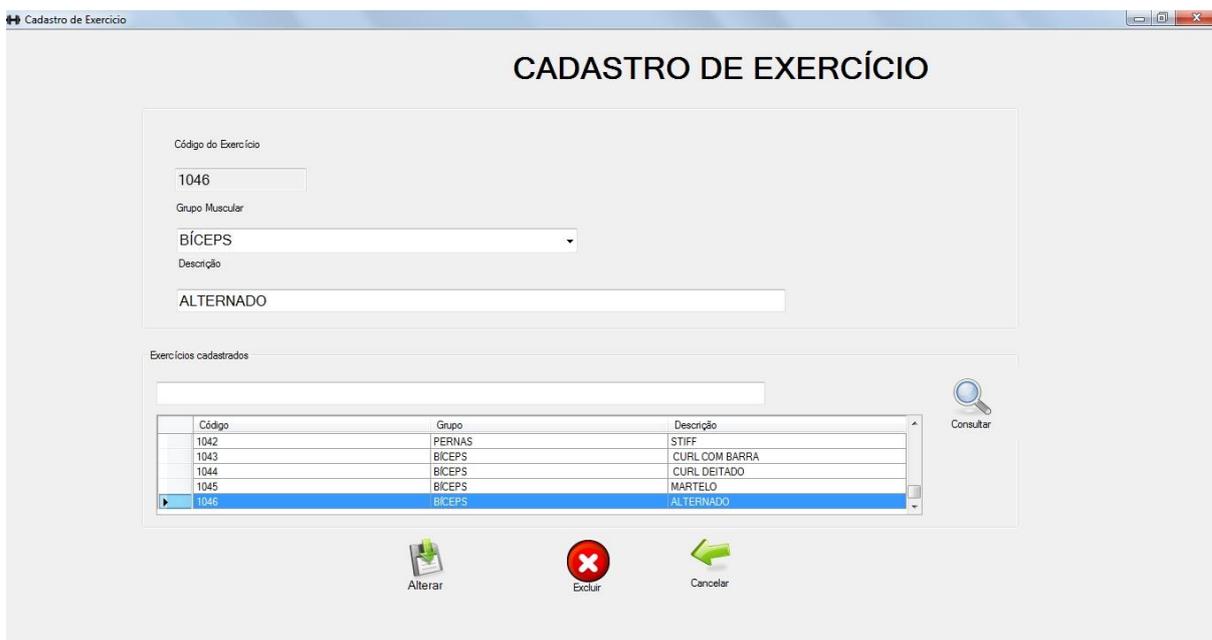


Figura 41 - Tela Cadastro de Exercícios

### Gerar Desempenho de Aluno

Registro nº

Aluno   Data

 Buscar aluno

Medidas

Altura <input type="text" value="1.75"/>	Peso <input type="text" value="70"/>	Cintura <input type="text" value="70"/>	Coxas <input type="text" value="50"/>
Tórax <input type="text" value="80"/>	Abdômen <input type="text" value="65"/>	Pantunilha <input type="text" value="50"/>	

Cálculo de IMC

Resultado   
Parabéns, seu IMC está normal para sua altura.

Índice de Gordura Corporal

Resultado  % de gordura  
Sua massa magra em quilos é de aproximadamente: 66,69 kilos

**Figura 42 - Tela Cadastro de Desempenho**

## CADASTRO DE INSTRUTORES

Home **Localizar**

Matrícula

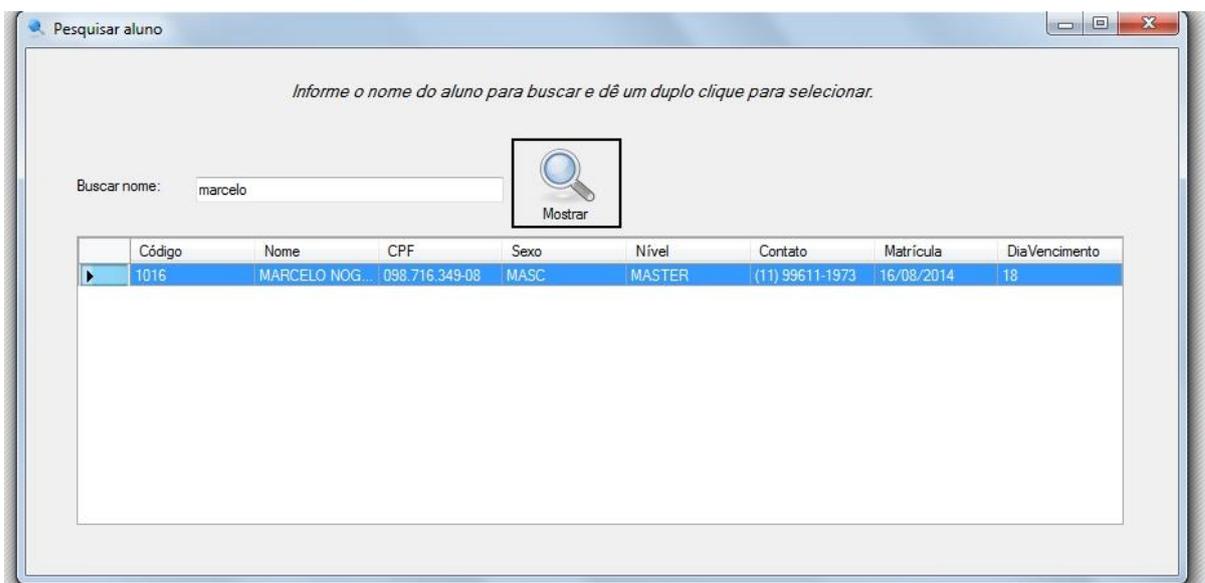
Nome

Data de admissão  Contato

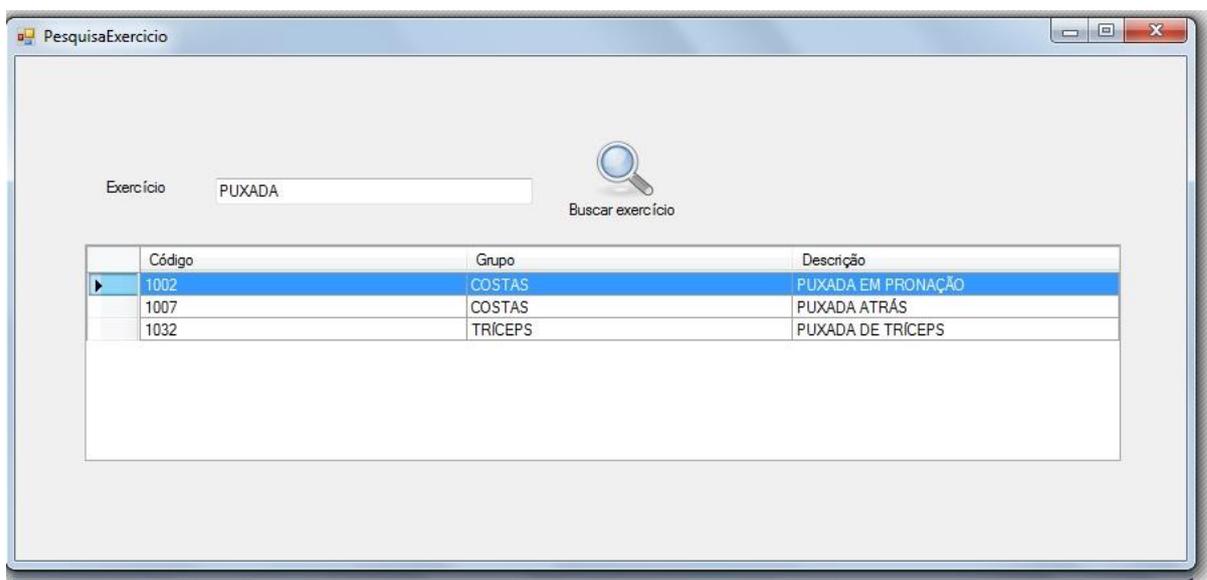
Senha  Confirmar Senha

**Figura 43 - Tela Cadastro de Instrutores**



**Figura 44 - Tela Pesquisar Alunos**



**Figura 45 - Tela Pesquisar Exercício**

CadastroTreino

## TREINOS

Cadastro de treinos

Código Treino  Código  JOÃO GUILHERME SILVA 

Instrutor

Tipo de Treino

Cadastro de plano de treino

Aluno     Séries  Repetições  Carga  Kg 

Carga	Exercício	Séries	Repetições
-------	-----------	--------	------------

**Figura 46 - Tela Cadastro de Treinos**

## 5. CONCLUSÃO FINAL

O planejamento inicial do projeto foi essencial para que o seu desenvolvimento pudesse ser realizado e assim atender os requisitos levantados, visando atingir a demanda do público-alvo.

Mesmo após a análise e planejamento, durante o desenvolvimento do projeto viu-se a oportunidade de inserir novas ideias ao projeto, bem como a reutilização de ideias consideradas inviáveis inicialmente. Com isso, o estudo das possibilidades e oportunidades do projeto deve estar sempre em constante evolução a fim de que não haja divergências durante o desenvolvimento do projeto.

Através dos requisitos levantados e dos diagramas apresentados, foi possível a percepção de que o sistema desenvolvido tem a capacidade de suprir as necessidades no gerenciamento de uma academia personalizada, tendo em vista que o software visa atingir estabelecimentos onde os fluxos de alunos são menores em relação às demais academias do ramo.

A utilização de ferramentas que auxiliam no desenvolvimento e que, ao mesmo tempo, facilitam a programação do projeto, foi de extrema importância.

Mesmo com o projeto dividido em etapas, conforme apresenta a Estrutura Analítica do Projeto (EAP) (figura 1, p. 20 deste projeto), ao término de cada etapa dava-se início à outra etapa, porém a etapa anterior não estava descartada de alterações ou melhorias, garantindo assim um frequente processo de evolução.

## REFERÊNCIAS

BEZEERA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 2ª Ed. – Rio de Janeiro: Editora Campus, 2007, 369 páginas.

CALÇADO, P. **Arquitetura de Camadas em Java EE**. Mundo Java. Rio de Janeiro, v.3, n.15, p.34-43, 2005.

FOWLER, Martin – **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos** / Martin Fowler, trad. João Tortello. – 3. Ed – Porto Alegre: Bookman, 2005

HEUSER, Carlos Alberto. **Projeto de banco de dados**. - 6ªed. - Porto Alegre,: Bookman, 2009

MARTINS, José Carlos Cordeiro. **Técnicas para gerenciamento de projetos de software**. Rio de Janeiro: Brasport, 2007

MILANI, A. MySQL: **Guia do Programador**. São Paulo: Novatec, 2007. 400 p.

MIRANDA, J. ¿**Salud, forma física, estética, bienestar?** ¿**Que lleva al usuario al gimnasio?** Revista Apunts, Educació Física i Esports, Barcelona; n. 26, p. 61-70, 1991.

NEVES, Leandro G.S. **A Importância Do Planejamento Para O Cumprimento De Prazos E Redução De Custos Em Projetos Na Área De Ti**. Revista E-Tech: Tecnologias para Competitividade Industrial. V.6, N.1, p.79-97, Florianópolis, 2013.

Disponível em:  
<http://revista.ctai.senai.br/index.php/edicao01/article/viewFile/332/293>. Acesso em:  
25 Fev. 2014

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de Informações Gerenciais: Estratégicas Táticas Operacionais**. 12ª Ed. – São Paulo: Editora Atlas, 2008, 299 páginas.

STADZISZ, Paulo César. **Projeto de Software usando a UML**. Centro Federal de Educação Tecnológica do Paraná, Departamento Acadêmico de Informática – Paraná: CEFET-PR, 2002. Disponível em  
<http://www.etelg.com.br/paginaete/downloads/informatica/apostila2uml.pdf>. Acesso em 01 Mar. 2014.